Wireless Static Hand Gesture Recognition with Accelerometers - The Acceleration Sensing Glove

Seth Hollar, John Kangchun Perng, Kristofer S. J. Pister
shollar@bsac.eecs.berkeley.edu
Berkeley Sensor & Actuator Center
University of California, Berkeley
497 Cory Hall
Berkeley, CA 94720
(510) 643-2236

Abstract

A glove with six 2-axis accelerometers on the finger tips and back of the hand is demonstrated using commercial-off-the-shelf components. With an RF transmitter, the glove can act as a wireless input device to a computer. With gravity-induced acceleration offsets, we have developed a text editor where each hand gesture refers to a letter of the alphabet. Twenty-eight static hand gestures are recognizable with acquisition times of up to 1 characters/second. The glove is a prototype device for cubic millimeter systems for application on the fingernails of the user.

Keywords

Wearable input device, hand-gesture recognition, data glove, human computer interaction, mouse pointer, etc.

Introduction

While mobile computing is becoming more realistic (for example PDA's), desktops are still the preferred choice of use. The keyboard offers the most efficient means for a user to input information into a computer. Our group developed an input device that both has the potential to offer equivalent input bandwidth and can be used in a mobile environment.

In particular, our goal was to demonstrate that accelerometers can be used to effectively translate finger and hand gestures into computer interpreted signals. To this end we developed the Acceleration Sensing Glove (ASG) that communicated using a wireless link to a desktop computer. Since first being introduced in ISWC [9], the glove has appeared in several magazines [1,2,3,4]. The glove uses six 2-axis accelerometers on the fingers and back of the hand. Using gravity-induced acceleration offsets, a program was developed to translate gestures to characters. The program emulated the use of a reduced keyboard, allowing the user to input any of the 26 letters of the alphabet with static hand gestures.

Accelerometers have previously been used on fingers; Fukumoto [8] demonstrated that accelerometers can be used to detect when fingers hit a hard surface. The ASG, however, does not require tapping on a surface to be used a gesture recognition device.

Hardware

The hardware consists of a wrist controller and six accelerometers, five on the fingertips and one on the back of the hand (see Figure 1). Each accelerometer(1.3x1.4cm), an Analog Devices ADXL202, contains 2 axes of measurement and had a range of with +/- 2g [5]. Wires send signals from the accelerometers to the forearm controller. An Atmel AVR AT90LS8535 microcontroller on the forearm controller (4.4x6.6 cm) converts the analog signal from the accelerometer to a 10 bit digital signal. The wrist controller transmits the sensor data via an RF link (RFM RX2000) with a carrier frequency of 916.5 MHz. We use a quarter wavelength antenna. While the raw data rate of the transmitter is 10 kbps, we implemented a packet layer with manchester encoding, a 16 bit flag, and an 8 bit CRC to ensure valid signal detection. 2(16 + 8 + 10 bit

* 12 signals) = 288 bits/ packet. Each packet is then transmitted at (10 kbps) / (288 bits / packet) = 34.7 packets/s or 34.7 samples/s for each channel. The overall power consumption of the glove was 38.4 mW at 3.2 Volts. RF transmission has been tested indoors at ranges over 4 meters with data rates not falling below 30 samples/s.

In the current version of the ASG, the electronics are powered by an external power supply, the HP E3630A. The glove also operates via two AAA batteries, but for reliable long term accuracy, the glove requires a power supply whose voltage does not change over time.

An RF receiver (RFM RX2010) and Atmel AT90LS8535 receive and decode the signal and send the data over the serial port to a computer. Algorithms on the computer interpret the hand motions.

Calibration

For static gesture recognition the accelerometer data is calibrated and filtered. The accelerometers can measure the magnitude and direction of gravity in addition to movement induced acceleration. In order to calibrate the accelerometers, we rotate the device's sensitive axis with respect to gravity and use the resultant signal as an absolute measurement.

To reduce high frequency noise from the sensors, we took a running average.

Lastly, we performed a coordinate transformation on the two axis accelerometers, transforming the raw x, y accelerations to θ and φ.

When the glove is not moving, i.e. in static situations, the only force acting on the accelerometers is gravity (Figure 3). The resulting vector from the projection of the gravity vector, G, into the plane defined by the 2 axis accelerometer is $\underline{R}_{acc}$. The orientation of the accelerometer relative to G is θ, with the acceleration plane offset from G by φ. The equations governing the transformation are described below:

$$\theta_j = \arctan\left(\frac{a_j^x}{a_j^y}\right) \tag{1}$$

where a_j^x, a_j^y are the accelerations along the x and y axes of the j^{th} accelerometer, respectively.

$$a_j^R = (a_j^x)^2 + (a_j^y)^2 \tag{2}$$

where a_j^R is the acceleration in the plane of the j^{th} accelerometer.

$$\varphi_j = \arcsin\left(\frac{a_j^R}{G}\right) \tag{3}$$

where G is the magnitude of gravity.

As a matter of convenience, I've defined η_i as a way of consolidating θ and φ.

$$\eta_i = \theta_i \quad \text{if } 0 \leq i < N \text{ and} \tag{4}$$

$$\eta_i = \varphi_{i-N} \quad \text{if } N \leq i < 2N \text{ where N is the total number of accelerometers (in this case, 6)}$$

Text Editor

We wrote a text editor program that allowed the user to input and edit characters on the computer with the ASG. The GUI program was developed with Microsoft's C++ Visual Studio 6.0. Figure 2 shows a cropped version of the program.

Text entry is a two step process:

1) Gesture learning
2) Gesture recognition

In the gesture learning stage, the idea is to sample gestures and store the data in a condensed form. For example, we sample n times for the gesture G_p. For each sample taken, twelve acceleration parameters come from the glove, namely η_i for i = 0 to 11. Instead of storing all 12 variables for each sampled gesture which amounts to 12n values, we calculate the mean and variance for each η_i.

Successive samples can be obtained for a more accurate assessment of the mean and variance. The equations below describe how the program determines the n^{th} iteration of the mean and variance.

Let $\eta_i(n)$ be the n^{th} sample of the variable η_i.

Let $\Delta(a, b) = a - b + 2\pi k$ where k is an integer and is chosen such that $-\pi < \Delta((a, b) \le \pi)$ (5)

$$\bar{\eta}_i(n) = \bar{\eta}_i(n-1) + \frac{\Delta(\eta_i(n), \bar{\eta}_i(n-1))}{n} \quad (6)$$

where $\bar{\eta}_i(n)$ is the mean of the set $\{\eta_i(1), \eta_i(2), \eta_i(3), \ldots \eta_i(n)\}$

$$\sigma_i^2(n) = \frac{(n-2)}{(n-1)}\sigma_i^2(n-1) + \Delta(\eta_i(n), \bar{\eta}_i(n-1)) \quad (7)$$

where $\sigma_i^2(n)$ is the variance of the set $\{\eta_i(1), \eta_i(2), \eta_i(3), \ldots \eta_i(n)\}$

In the gesture recognition stage, runtime η_i are used to determine whether a gesture is matched. Because there are twelve parameters for a given gesture, G_p, we use the variance to determine the relative importance of each η_i. In particular, we use the mean and variance calculated in the gesture learning stage to create a function that calculates the probability that G_p is matched. While the mean determines the absolute error of a sample, the standard deviation is used to calculate the relative weight of each parameter.

Assuming that a sampled variable, η_j, is gaussian in nature and uncorrelated to other variables, we can write the probability distribution function for a particular gesture as:

$$P(\eta_i) = \frac{1}{\sigma_i^p\sqrt{2\pi}}\exp\left(-\frac{(\eta_i - \bar{\eta}_i^p)^2}{2(\sigma_i^p)^2}\right)$$ where $\bar{\eta}_i^p$ and σ_i^p are the mean and standard deviation, respectively, of η_i for gesture G_p. (8)

In order to determine the probability that an incoming variable lies outside a given region, we need to integrate the PDF.

Applying the transform $z_i^p = \frac{|\eta_i - \bar{\eta}_i^p|}{\sigma_i^p}$, $C(z_i^p)$ represents the probability that a sampled variable lies outside the region $(\bar{\eta}_i^p - |\eta_i - \bar{\eta}_i^p|, \bar{\eta}_i^p + |\eta_i - \bar{\eta}_i^p|)$. Mathematically, this can be described as:

$$C(z_i^p) = \frac{2}{\sqrt{2\pi}}\int_{z_i^p}^{\infty} e^{-\frac{x^2}{2}}dx = \operatorname{erfc}\left(\frac{z_i^p}{\sqrt{2}}\right) \quad (9)$$

In our experiments, we scaled our standard deviations to yield higher probabilities for $C(z_i^p)$. Therefore, (9) is rewritten with a scaling factor of K:

$$C(z_i^p) = \operatorname{erfc}\left(\frac{z_i^p}{K\sqrt{2}}\right) \quad (10)$$

We set K = 2 for all experiments mentioned in this paper.

$C(z_i^p)$ indicates the chance that data from the glove matches G_p for a particular η_i. For example, if z_i^p is relatively small, say 0.1 then $C(z_i^p)$ is 92%. The η have a high chance of matching G_p. Likewise if z_i^p is large, $C(z_i^p)$ approaches zero indicating that there is a very small chance that the η match G_p.

There are twelves variables, η_j, that in turn create twelve probabilities, C_j. To calculate the probability that a single gesture is matched, we simply multiply the probabilities for each η_j. Namely:

$$\mathrm{Prob}(G_p) = \prod_i C(z_i^p) \tag{11}$$

The algorithm computes $\mathrm{Prob}(G_p)$ for each learned gesture G_p and interprets the signal as gesture $G_{p'}$ where p' is:

$$p' = \operatorname{argmax}(\mathrm{Prob}(G_p)) \text{ where } p' \in p \tag{12}$$

Determining When to Recognize New Gestures

Because the gravity vector can only be deciphered when the accelerometers are not moving, the text editor program is designed to recognize only static gestures. Furthermore, once a static gesture is recognize, the program needs a way to know when to look for a new gesture. This is done by detecting when the hand has moved. Once significant movement is detected the program prepares to detect a new gesture.

To do this, we implemented a running average. The program calculates the mean of the last P samples in time for the $a_{x,j}$ and $a_{y,j}$.

$$\bar{a}_{x,j}(k) = \frac{1}{P}\sum_{i=k-P+1}^{k} a_{x,j}(k)\ ,\ \bar{a}_{y,j}(k) = \frac{1}{P}\sum_{i=k-P+1}^{k} a_{y,j}(k) \tag{13}$$

Motion is detected by looking at the difference of the present values, a, and the running average values, $\bar{a}$. The difference is then squared and summed over all variables and the last P samples in time.

$$\Lambda(k) = \sum_{j=N}\left(\sum_{i=k-P+1}^{k}(a_{x,j}(i)-\bar{a}_{x,j}(i))^2 + (a_{y,j}(i)-\bar{a}_{y,j}(i))^2\right) \tag{14}$$

Threshold comparisons on Λ can determine when and when not to look for new gestures.

Results

For the gesture learning stage, multiple samples were taken for each gesture. On average about 28 samples for each gesture were taken to ensure a high chance of matching. Further samples were taken on gestures if recognition was unreliable.

The table below shows the mean of the standard deviation for the sampled gestures taken over all gestures.

Table 1: Mean of Standard Deviation for Sampled Gestures

	Index Finger	Ring Finger	Pinky Finger	Back of Hand	Middle Finger	Thumb
σ for θ_j	8.9°	10°	10°	21°	9.0°	14°
σ for φ_j	6.2°	7.0°	7.1°	5.6°	4.9°	6.5°

In the recognition stage, the text editor recognized 28 gestures in all (see Figure 6), 26 letters of the alphabet, space bar, and a delete gesture. We ran an experiment with a single user. After a few hours of practice, the user was able to go through the alphabet reliably with an average character recognition speed of 0.91 characters/sec. 162 characters were consecutively recognized on a single trial. The results match favorable with other alternative input devices. Thomas et. al demonstrate the Kordic keypad can input characters at a rate of 1.25 characters/s [6].

Table 2: Character Recognition

Total Recognizable Gestures	Mean Character/second	Std. Deviation of Character/ second	Number of Consecutive Gestures Recognized
28	0.91	0.04	162

We chose hand gestures we felt the ASG could distinguish between. While most static hand gestures can be recognized, the glove cannot recognize all hand gestures. For example, not all gestures from the American Sign Language can be recognized since individual accelerometers cannot distinguish rotations about the gravity vector. As in Figure 4, the two hand gestures are different, but cannot be differentiated. The middle finger is in a new position, but is only a rotation about the gravity vector.

To first order, we calculated the theoretical limit of the total number of static hand gestures that can be distinguished by the ASG. We took into account both the constraints imposed by the hand and finger kinematics and the ability of the user to keep ones gesture still. Taking into account the data from Table 1 and assuming a minimum of three standard deviations between gestures, we calculated a theoretical limit of 4000 static hand gestures.

While with the current design of the ASG, character speeds have not exceeded 1 character/second, a number of improvements on future designs could significantly increase the input rate. The RF link limited sensor throughput to 34 samples/second. Both averaging and motion detection further limited the speed by which characters could be recognized. By increasing the data rate of the RF link, these barriers can be overcome.

Additionally, this analysis has solely concentrated on recognition of static hand gestures. Moving, or dynamic, hand gestures can also be recognized. Unfortunately, for reasons of simplicity, we did not pursue this route of investigation. Nevertheless, dynamic hand gesture recognition could dramatically increase both the number of distinguishable gestures and the speed of recognition.

Future Potential

The finger acceleration glove is a large-scale device used to model the functionality of Smart Dust [7] on fingers. Smart Dust is a project to integrate communications, intelligence, power, and sensors into a package no larger than 1 mm^3 (see Figure 5). Integrating a single chip wireless solution with a MEMS accelerometer would yield an autonomous device small enough to apply to the fingernails. Because of their small size and weight, these Smart Dust devices would be less noticeable than one's own eyeglasses, providing no more discomfort than fingernail polish. People would have instantaneous input access to the digital world at all times, facilitating a paradigm shift in human-computer interaction.

Some potential applications for the Acceleration Sensing Glove are: a wireless wearable mouse pointing device, a wireless wearable keyboard, hand motion and gesture recognition tools, virtual musical instruments, computer sporting games, and work training in a simulated environment.

References

[1] Flavia Caroppo, Massimo Chieruzzi, “Bit di futuro a portata di mano”, Newton, pp 113-118. April 2000.

[2] “Hand Job”, Wired Magazine, pp 57 January 2000.

[3] George Johnson, “Only connect from swarms of Smart Dust to secure collaborative zones, the omninet comes to you”, Wired Magazine, pp150 - 164, January 2000.
[4] Peter Weiss, “Smart Outfit”, Science News, pp 330-332, Vol. 156, November 20, 1999.
[5] Analog Devices, Norwood, MA, http://www.analog.com, September 1999.
[6] Bruce Thomas, Susan Tyerman, Karen Grimmer, “Evaluation of Three Input Mechanisms for Wearable Computers,” IEEE 1997 ???? pp 2-9
[7] http://robotics.eecs.berkeley.edu/ ~pister/SmartDust, September 1999.
[8] Fukumoto, M and Tonomura, Y “Body Coupled FingeRing: Wireless Wearable Keyboard”, CHI97 Conference Proceedings, pp147-154, Atlanta, March 1997.
[9] J. Perng, B. Fisher, S. Hollar, K.S.J. Pister, “Acceleration Sensing Glove,” ISWC International Symposium on Wearable Computers, San Francisco, October 18-19th, 1999.

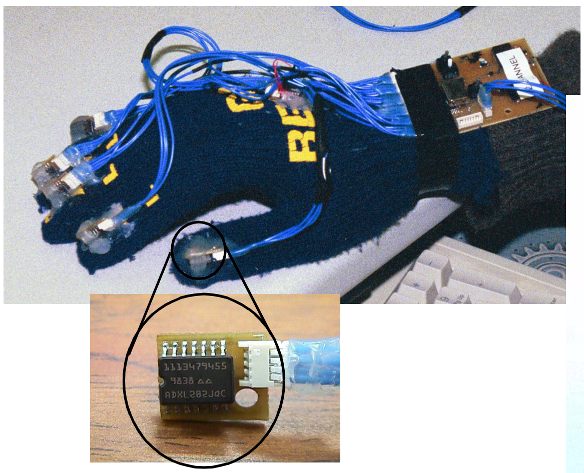

Fig. 1a Acceleration Sensing Glove Hardware. Accelerometers are attached to the fingertips and back of the hand. A wrist controller sends the accelerometer data wirelessly to a computer.

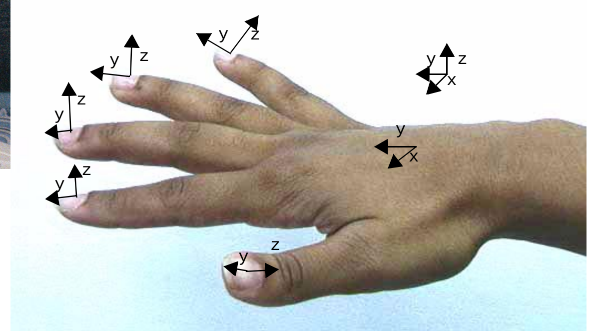

Fig. 1b Arrows on the hand show the location of accelerometers and their sensitive directions. Note that the sensitive direction of the accelerometer is in the plane of the hand.

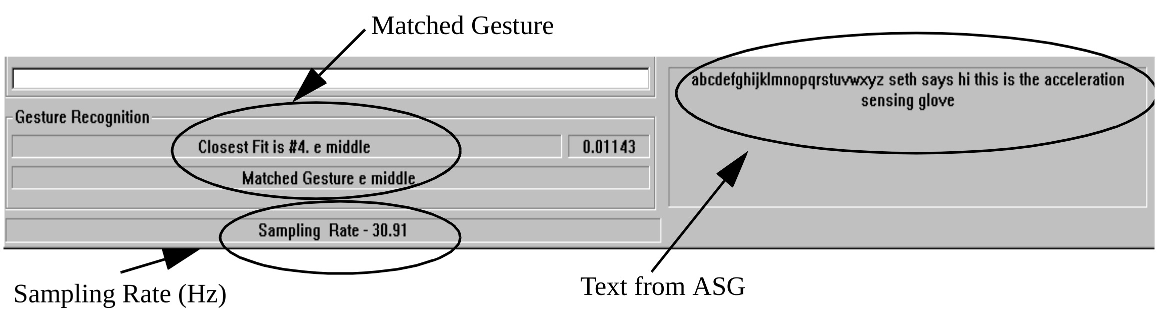

Fig. 2 Software for text editor. Software matches to gesture and outputs text on the left side of the window. Sampling rate is about 33 samples per second.

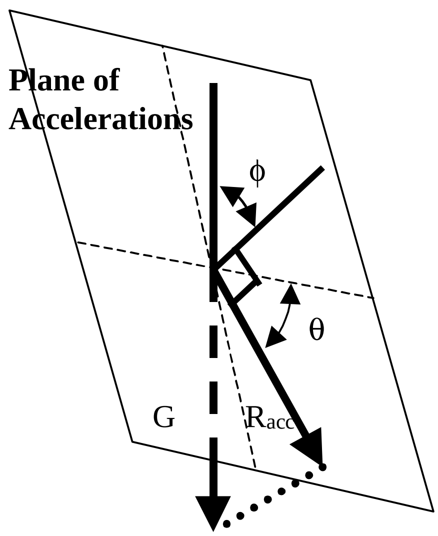

Fig. 3 Acceleration plane relative to gravity vector. Accelerations normal to acceleration plane cannot be detected by accelerometers.

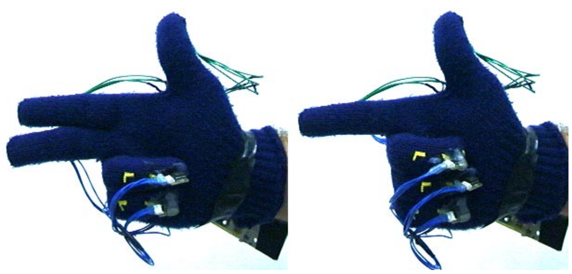

Fig. 4 Two gestures that cannot be distinguished with the Acceleration Sensing Glove. Middle finger is rotated about gravity vector. Middle finger's acceleration signal left unchanged between two gestures.

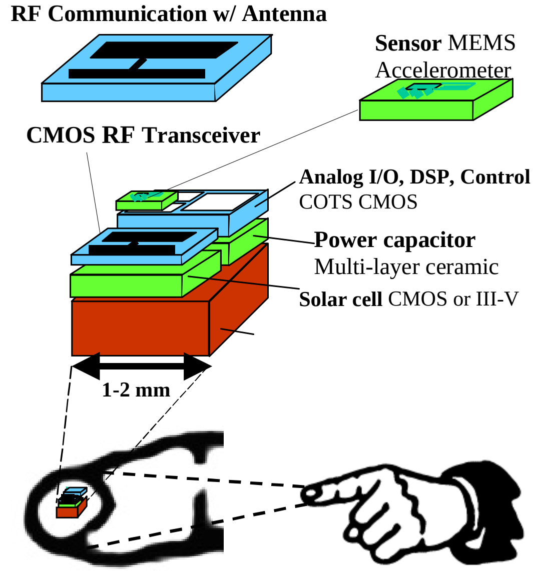

Fig. 5 Future goal of Smart Dust on a finger.

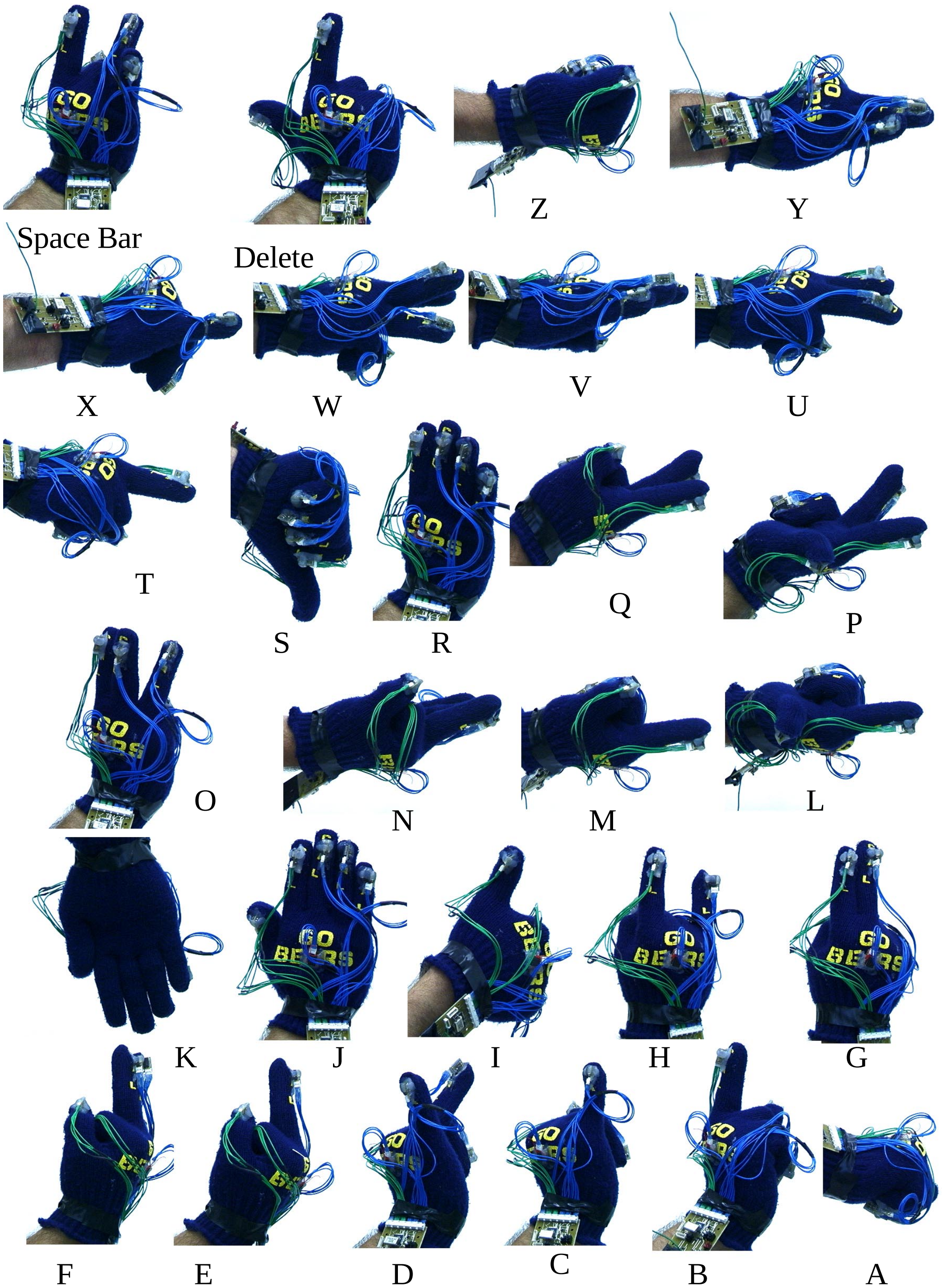

Fig. 6 The alphabet plus space bar and delete make up the twenty-eight gestures used in the text editor program.