Gesture Keyboard with a Machine Learning Requiring Only One Camera

Taichi Murase, Atsunori Moteki,Genta Suzuki, Takahiro Nakai, Nobuyuki Hara,Takahiro Matsuda
Fujitsu Laboratories Ltd.
1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
+81-44-754-2678
{murase.taichi | moteki.atsunori | suzuki.genta | t-nakai | hara.nobu | tmatsuda}@jp.fujitsu.com

ABSTRACT
In this paper, the authors propose a novel gesture-based virtual keyboard (Gesture Keyboard) that uses a standard QWERTY keyboard layout, and requires only one camera, and employs a machine learning technique. Gesture Keyboard tracks the user's fingers and recognizes finger motions to judge keys input in the horizontal direction. Real-Adaboost (Adaptive Boosting), a machine learning technique, uses HOG (Histograms of Oriented Gradients) features in an image of the user's hands to estimate keys in the depth direction. Each virtual key follows a corresponding finger, so it is possible to input characters at the user's preferred hand position even if the user displaces his hands while inputting data. Additionally, because Gesture Keyboard requires only one camera, keyboard-less devices can implement this system easily. We show the effectiveness of utilizing a machine learning technique for estimating depth.

Categories and Subject Descriptors
H.5.2 [**Information interfaces and presentation**]: User Interfaces – *Interaction styles, prototyping.*

General Terms
Design, Human Factors

Keywords
Hand Gesture, Keyboard, Gesture Recognition, Machine Learning,

1. INTRODUCTION
Recently, keyboard-less devices such as tablet PCs and smart phones have rapidly gained popularity. However, character-input interfaces for such devices have not yet achieved better usability than the standard physical QWERTY keyboard. Therefore, keyboard-less devices have not been used for generating documentation especially in business. People are expecting the development of a new character-input interface[3] [4].

The 1line keyboard[2] uses less keyboard space than a normal software-keyboard, so the user can use a wide display. However, the user must use a hand-flicking action to select words in addition to a normal input action. Furthermore, because display sizes of smart phones are smaller than tablet PCs, it seems difficult to input characters. VKey[6] which is a tabletop unit that projects a keyboard image onto any flat surface. However, VKey was also less convenient for users because it required users to use a fixed keyboard image. Additionally, VKey requires additional devices, e.g. a projector, which limits the application of such technique on keyboard-less devices. A previous version of Gesture Keyboard[5] which did not implement a machine learning technique that was proposed by the authors. That calculated what key in the depth direction was input by using only figure features of the user's hands. However, figures vary appreciably between individual users. This proposes a problem in calculating parameters. To solve the above problems, the authors propose a gesture-based virtual keyboard (Gesture Keyboard) that has the standard QWERTY keyboard layout, and requires only one camera, and employs a machine learning technique.

AH'12, March 08–09, 2012, Megève, France.

2. PROPOSED GESTURE KEYBORAD
Gesture Keyboard captures the user's hands and fingers using one camera that is setup in front of the user at an upright position on the desk (Figure 1). Furthermore, Gesture Keyboard detects the user's fingers, estimates the depth, and determines whether data was typed. One difference from the previous Gesture Keyboard is that a machine learning technique, Real-Adaboost that adopts HOG features of the user's hands image, is used for estimating depth and improving performance.

Figure 1. Prototype implementation.

Each virtual key follows a corresponding finger. Therefore, it is possible to input keys at the user's preferred hand positions even if the user displaces his hands while inputting data. The display of Gesture Keyboard is shown in Figure 2. Because it requires only one camera, even keyboard-less devices which are more compact and thinner can implement this feature easily.

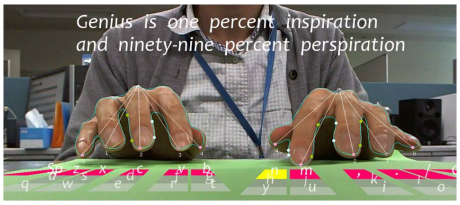

Figure 2. Gesture Keyboard display.

3. PROTOTYPE IMPLEMENTATION

The prototype implementation uses a convertible laptop PC and a web camera (size: 320 x 180). The authors assume that a tablet PC has been installed with Gesture Keyboard, and the camera is positioned under the monitor (Figure 1). Next, the paper will introduce Gesture Keyboard operating procedures.

3.1 CALIBRATION

System calibration consists of the following two steps: First Gesture Keyboard registers the color of the user's hand, detects the hand region on YUV color space using color information [1], and registers the features (e.g. centroid, width, height, finger width) of the hand as the home row. Second, the user touches an area of the table surface, and the vertical position is registered for determining a key has been typed.

3.2 EVERY FRAME PROCEDURE

This procedure consists of detecting the user's fingers, determining whether a key has been typed, and estimating the depth.

First Gesture Keyboard detects the user's fingers using the contours of the hands and reference features registered during calibration. Because the image is two dimensional, one finger is erroneously detected if one finger overlaps another one, but Gesture Keyboard also takes such situations in to consideration in this process.

Next Gesture Keyboard determines whether data has been typed using the estimated fingers. If the tip of a finger crosses a vertical position of the table surface that was registered during calibration, Gesture Keyboard determines that a key has been pressed. Because the camera is set parallel to the table surface, Gesture Keyboard detects vertical movements with relative ease.

The last step is to estimate hand depth. This is a major problem to overcome because the camera only recognizes in two dimensions. However, the authors attempted to solve this problem by implementing Real-Adaboost which is a machine learning technique that adopts HOG features of user's hands image.

Gesture Keyboard calculates HOG features using only the hands region image when the user types (Figure 3). The hand region image is normalized to 160 x 180pixels. The right hand image is mirrored to orient the left hand image. The number of gradient direction is 8 per 45degrees (Figure 3).

Three classifiers of Real-Adaboost are connected in parallel, which correspond the three rows of an alphabet QWERTY keyboard. The classifier of highest likelihood is selected as the row of input.

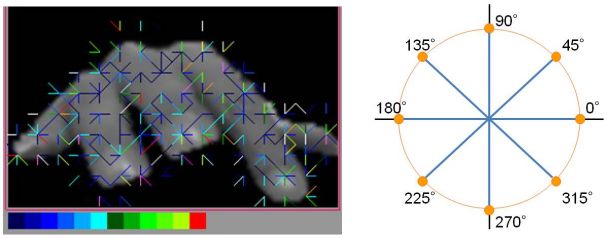

Figure 3. HOG features of a hand image.

4. EXPERIMENTS

The authours evaluated the character input error rate using the proposed Gesture Keyboard and previous one. One examinee input 690 charcters(69 characters of two pangrams ten times) without regard to speed. The parameters were that cell-size was 10x10, number of learning was 50 times, and number of learning sample was 787 samples.

Table 1 presents the result. *w/ learning* means a proposal Gesture Keyboard, *w/o learning* means a previous one. The vertical error decreased of 31.3% (from 32 to 22) by using a machine learning.

Table 1. Character input error rate

Error	w/ learning	w/o learning
Total	5.9% (39)	7.4% (51)
Depth	3.2% (22)	4.6% (32)
Horizontal	2.0% (14)	
False Input	0.4% (3)	
Miss Input	0.3% (2)	

5. CONCLUSION AND FUTURE WORK

The authors developed a novel gesture-based virtual keyboard (Gesture Keyboard) that adopts the standard QWERTY keyboard layout and requires only one camera with a machine learning technique. The depth character input error decreased of about 31.3% by utilizing a machine learning, which is not yet adequately low. Furthermore, the authors particularly have not implemented thorough evaluations to provide a quantitative assessment of the system's robustness. Therefore, the authors still have many future tasks, especially including efforts to improve the system's robustness for detecting the hand region independent of room lighting and user differences, and to improve the accuracy of depth estimation. The authors will work to solve these issues using different features and learning methods.

6. REFERENCES

[1] Emi Tamaki, et al. A Robust and Accurate 3D Hand Posture Estimation Method for Interactive Systems. Information Processing Society of Japan 2010.

[2] Frank Chun Yat Li, et al. The 1Line Keyboard: A QWERTY Layout in a Single Line. UIST '11. ACM, Santa Barbara, CA, USA

[3] Juan Pablo Wachs, et al. Vision-Based Hand-Gesture Applications in communications of the ACM (February 2011, vol.54, No2)

[4] Mathias Kolsch, et al. Keyboards without Keyboards: A Survey of Virtual Keyboards. UCSB technical Report 2002-21, July 12, 2002

[5] Taich Murase, et al. Gesture Keyboard Requiring Only One Camera. UIST '11. ACM, Santa Barbara, CA, USA

[6] Vkey: Virtual Devices, Inc., http://www.virtualdevices.net/index2.htm