Technology Reports

User Operations Assistance | UI | Proximity Position Information

User Operations Assistance Functions Using Finger Proximity Position Information

The ability to operate Smartphones intuitively through the use of a touch panel is one of the notable features of Smartphones, but because the touch panel does not allow for "focus" on objects of operation before they are touched, it is difficult for the user to confirm the process about to take place before actually touching an icon. To address this issue, NTT DOCOMO developed an assist function that detects finger proximity position and makes use of that information, and incorporated this function in its 2013 summer model. This article explain how the finger proximity position is detected and how the user operation assistance function featured in the summer 2013 model works by taking a close look at the operation of the software keyboard in particular.

Communication Device Development Department ***Masashi Tagaya***
Takashi Okada
Research Laboratories ***Sadanori Aoyagi***

1. Introduction

The smartphone has become the dominant mobile terminal in the market in recent years, and its operability is vastly different from conventional feature phones. Whereas a feature phone has a separate display and hardware keyboard for performing operations, the smartphone has an integrated structure that combines the display and operation components, so that users directly touch icons displayed on the screen to perform operations. Therefore, smartphone users can operate applications and manage data intuitively. Unlike feature phones, however, smartphones do not provide for "focus*1" on the operation target, so it is difficult for users to confirm details of processes before they actually select an icon (**Figure 1**).

Focus is generally performed using the hardware keys (4-way keys) on a feature phone, or a mouse or stylus (touch pen etc.). To provide the same focus in a smartphone, however, would not only require an additional device but would also result in the loss of the benefits of intuitive operability achieved by directly touching the icons displayed. To overcome this drawback, it was our view that focus using finger proximity position information would offer an effective solution for enabling focus while retaining the basic smartphone "touch with the finger" operability.

With this in mind, we developed a user operation assistance function that uses finger proximity position information and incorporated it into our 2013 summer model. Using a touch screen that obtains the coordinates of the user's

*1 **Focus**: Highlighting an icon etc. to confirm a process before inputting or executing.

finger even when the finger is not actually touching the screen but just over it, this function causes the focus to move in response to finger coordinates and to display additional information to assist the user in confirming selections. Thus, this function improves user operability and convenience without sacrificing the advantages of smartphone intuitive operations.

This article describes a user operation assistance function we developed using finger proximity position information technology.

2. Obtaining Finger Proximity Position Information

As shown in **Table 1**, there are three main methods of obtaining finger proximity position information. For this development we adopted an electrostatic capacitance touch screen for the following reasons: to obtain finger proximity position information across the entire touch screen, ensure stable operation unaffected by light, etc. in the vicinity of the phone and, above all, eliminate the need for an additional device.

Being more sensitive than ordinary touch screens, this touch screen can detect tiny changes in electrostatic capacitance to identify finger coordinates even when the finger is approximately 1 cm away from the touch panel. This enables advanced focus to be displayed for objects and application icons, etc. by obtaining the position that the user is about to touch (**Figure 2**). Furthermore, the Application Programming Interface (API)*2 for obtaining finger proximity position information is supplied as standard with Android™*3 4.0, and applications can be developed using this.

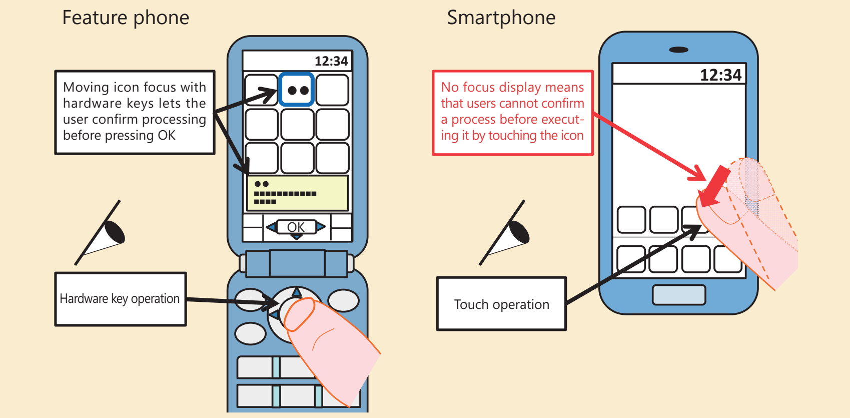


Figure 1 Difference in feature phone and smartphone operation

Table 1 Comparison of methods for obtaining proximity coordinates

	Ultrasonic sensor method	Optical sensor-in-cell touch screen method	Electrostatic capacitance touch screen method
Coordinate acquisition method	Ultrasonic waves come into contact with the finger, and coordinates are measured from the reflected waves	Coordinates determined from received light intensity by optical sensors positioned in display cells	Coordinates determined by changes in electrostatic capacitance on the touch screen
Detection area	Ultrasonic directivity (NA)	Entire touch panel surface (G)	Entire touch screen surface (G)
Detection distance	5 cm or more (G)	Susceptible to light in the vicinity (NA)	Approx. 1 cm (A)
Additional device	Requires ultrasonic transmitter and several receivers (NA)	No additional device needed with existing equipment (G)	No additional device needed with existing equipment (G)

G (Good), A (Applicable), NA (Not Applicable): Indicate suitability for this development.

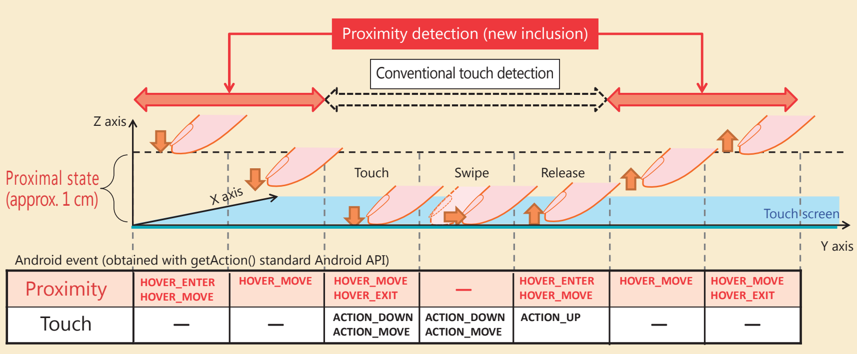


Figure 2 Difference between finger proximity detection (new inclusion) and touch detection

*2 **API**: An interface that makes the functions provided by the OS, middleware and other such software available to upper-level software.

*3 **Android™**: A trademark or registered trademark of Google, Inc., in the United States.

This API is the standard getAction () method that provides similar mechanisms for obtaining touch detection and proximity information, through which data can be obtained as separate distinct events as ACTION_MOVE for touch detection and HOVER_MOVE for proximity detection. The use of this standard API therefore enables not only pre-installed applications but also any application to be operated with finger proximity.

On the other hand, increased touch screen sensitivity makes the screen more susceptible to electromagnetic noise, etc. from internal mobile terminal components such as the display, which can increase touch screen jitter*4. When touch screen jitter is significant, focus may erroneously be applied to unintended screen objects, which can conversely impair operability.

To resolve this issue, we decided to install a proximity detector filter onto the touch screen output. Using results of surveys of the touch screen jitter characteristics, smartphone manufacturers have optimized the filter constant to combine both jitter suppression and tracking capability, and introduced position adjustment to decrease deviations between intended position and detected position.

To assess the filter, a touch screen evaluation robot similar to the one shown in **Photo 1** was used.

Using this device, the same straight-line and curved motions as well as complex movements that users make when selecting an object were repeatedly performed. Then, various parameters such as detection distance, detection position, and focus tracking were adjusted to optimize the touch screen performance for the user experience. Bearing in mind user scenarios apt to occur in general use, touch screen performance with water droplets on the screen, LCD protective film attached, and sebum on the screen were also assessed to confirm that these had no adverse effects on operability.

3. Developing the Application

As shown in **Figure 3**, the assistance function that uses finger proximity position information in the model released in the summer of 2013 was designed to support a number of applications including pallet UI (standby/home screen), software keyboard, phone unlock functions with pattern input and browsing.

This chapter describes how we improved operability using finger proximity position information in frequent input situations such as e-mail creation using the software keyboard – the input method most likely to significantly affect operability.

3.1 Assistance Function with Finger Proximity Position Information

As shown in **Table 2**, there are three ways in which finger proximity position information detected with a touch screen can be used to assist user input operations.

The first is to use the information for guide display, in which related information or function icons of segments, etc. are displayed in easy-to-see positions, so that the relevant information is not hidden under the user's finger. The second

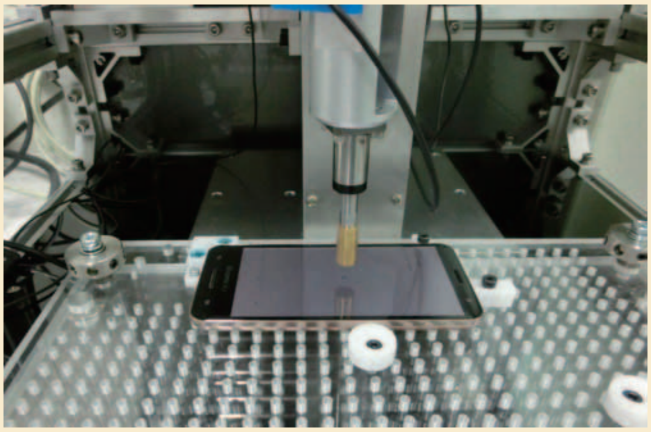


Photo 1 Evaluating touch screen jitter characteristics

*4 **Touch screen jitter**: A touch screen phenomenon where detected coordinates move in small increments even when the finger in proximity is motionless over the touch screen.

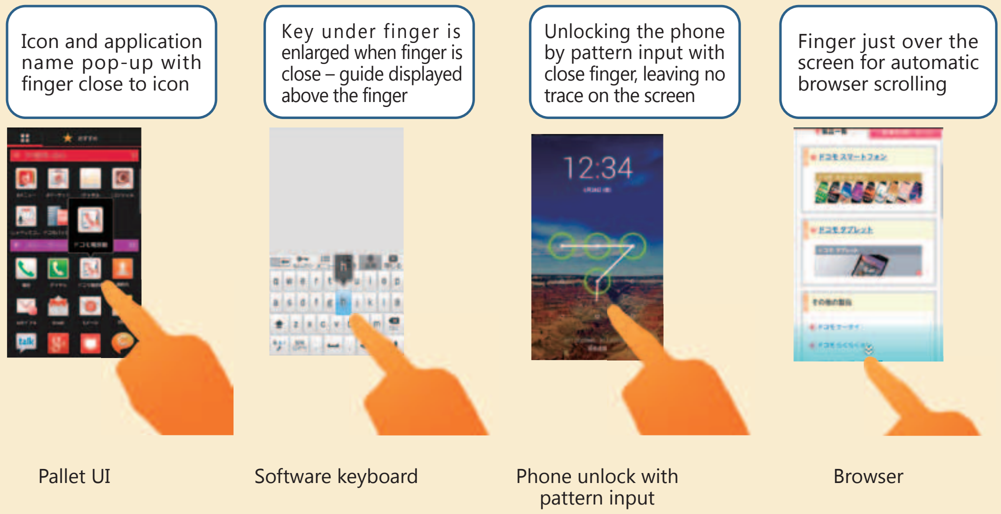
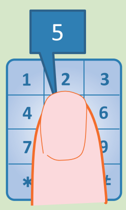
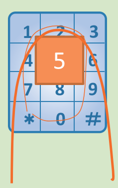
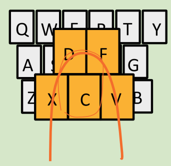


Figure 3 Assistance functions using finger proximity position information

Table 2 Types of input assistance using proximity coordinates

	Guide display	Touch detection area enlargement	enlarged display
Assistance details	Anticipates the part to be touched, and displays the data/function icon etc. in an easy-to-see position	Anticipates the part to be touched, and expands the detection area to absorb finger shake	Anticipates the part to be touched, and enlarges it to make it easy to see
Applicable area	Touched parts	Touched parts	A certain area around a touched part, including the touched part itself
Effective solution for:	Displaying objects or icons could be hidden by the finger	Operating objects or icons that are small on the display	Enlarging small display details temporarily to facilitate reading
Image			

is to expand the touch detection area in anticipation of the place to be touched and thus absorb finger trembling, making it easier for users to operate small display items. The third method is to enlarge the display in anticipation of the area to be touched while enlarging the surrounding area to make it easy to see. This is effective as temporary enlargement of detail display to facilitate reading with browsers, etc.

With smartphones, text is normally input by touching the QWERTY keys*5 or numeric keypad*6 (ex. 3 × 4) displayed. In this development, we studied the incorporation of software keyboards based on the three basic assistance functions described above.

3.2 QWERTY Key Input

Because QWERTY keyboards contain many keys, and each key occupies only a very small area, it is impossible to see the key that the user is pressing, because his or her finger hides the key. Both these factors make it very easy to make mistakes when inputting text.

Guide displays are an effective solution to the latter issue of hidden keys, and to solve the former issue, the two methods of enlarging the detection area and enlarging the display can be applied. Use of display enlargement approach can substantially impede keyboard visibility because key positions are gradually shifted from their original positions due to the repetition of the enlargement. To solve this issue, we used the detection area expansion

*5 **QWERTY keys**: A widely used keyboard layout for inputting the alphabet.

*6 **Numeric keypad**: A dedicated keypad for inputting numbers.

method (so that only one key is expanded on the display).

Based on the above considerations, we incorporated an assistance function with QWERTY key input that combines both a guide display and detection area enlargement. As shown in **Figure 4**, this means that when the finger is detected as being close, the key under the finger is inferred from the coordinates. The detection area of the key under the finger (and its display) is then enlarged, and the key under the finger is indicated with a guide display above the user's finger, enabling the user to identify the key more easily.

3.3 Numeric Keypad Input

Unlike QWERTY key input, numeric keypad input does not have the same small key area issues, and flicking*7 is generally used as the method of Japanese input, with a second operation required after touching. If the user is unfamiliar with the relationship of the characters assigned to directions (four characters assigned for up, down, left and right), however, input will be slowed because the user needs to first touch and confirm the direction with the flick guide display before flicking.

To resolve this issue, we incorporated a function that displays a flick guide when the finger is brought over a key as shown in **Figure 5**. The incorporation of this function enables the user to move smoothly from touching to flicking even when the user is not familiar with the keys assigned to flick directions and, therefore, to reduce input time.

4. Conclusion

This article described the user assist

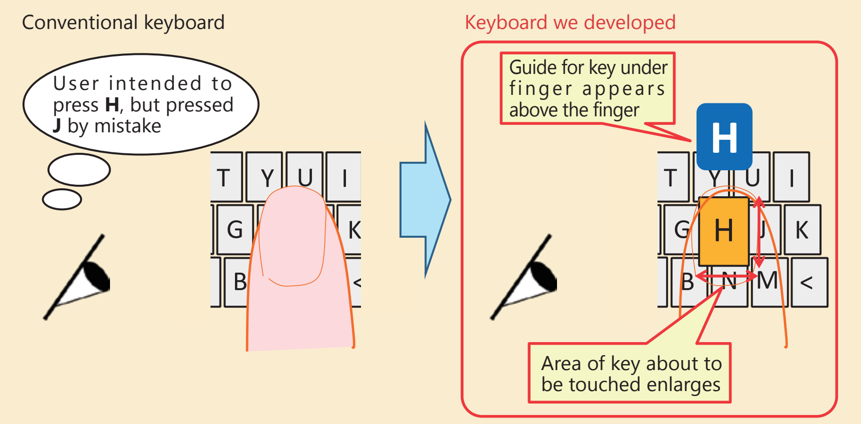


Figure 4 QWERTY key input

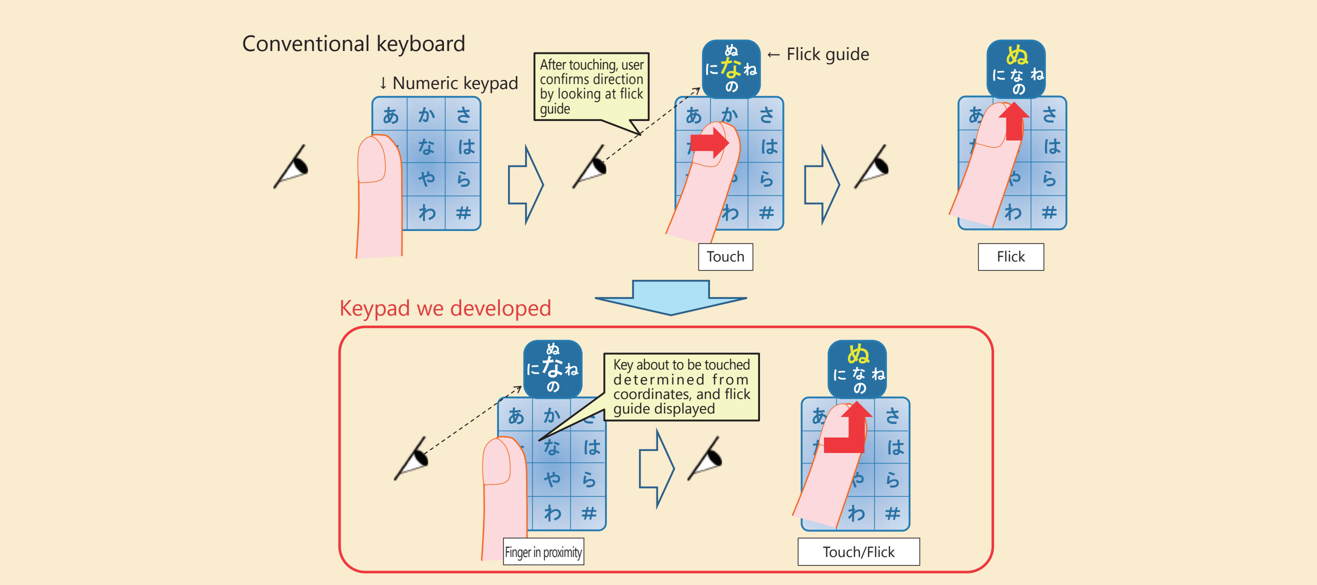


Figure 5 Numeric keypad input

*7 **Flick**: Method of input performed by rapidly moving a finger.

function incorporated in the 2013 summer smartphone model that uses finger proximity position information, a touch panel designed to obtain finger proximity position information, and a software keyboard as an example of an application incorporating these assistance functions.

We believe that the incorporation of this assist function in smartphones can shorten the input time of users unfamiliar with smartphones and reduce the number of input errors of experienced users. As a function incorporated into our smartphones for the first time in the summer of 2013, it may require improvement in the future depending on results of its usability. This article has described the function developed with an Android API, but it is also possible to obtain finger proximity position information with applications written in HTML 5*8 using HTML tags or JavaScript*9 methods.

In the future, we will play close attention to the development of technologies for improving the accuracy of obtaining touch panel proximity position information and make further improvements as required. We also intend to expand the application of this function to other areas such as games and entertainment, and to create ways of using this technology so that user interfaces with 3-dimensional intuitive operability become more widely available.

---

*8 **HTML5**: An enhanced version of HTML formulated by WHATWG and W3C.

*9 **JavaScript**: A script language appropriate for use in Web browsers. JavaScript is a registered trademark or trademark of Oracle Corporation, its subsidiaries and affiliates in the United States and other countries.