PZBoard: A Prediction-based Zooming Interface for Supporting Text Entry on a Mobile Device

Abstract
In this paper, we propose Predictive Zooming Keyboard (PZBoard) which predicts a target position from the finger movement above the touch-screen of a mobile device and that enlarges a part of the keyboard around the predicted position. We use a hover function of the mobile device to obtain the finger position above the touch-screen. While a finger is detected, a part of the keyboard around the finger position are enlarged. When the user moves his/her finger fast to enter a distant key, the target position is predicted and the center of enlargement moves to the predicted position. Using prediction, the system can start drawing the screen before the finger reaches the target position, which reduces the system’s latency and quickens the user’s response. The proposed interface does not force the user to perform additional operation for enlarging keys, and enables stable selection of both near and distant keys by fixing or changing the center of enlargement based on the velocity of the finger.

Author Keywords
Text entry; hover input; touch-screen; zooming interface

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces. – Input devices and strategies
**Introduction**
In recent years, mobile devices such as smartphones equipped with a touch panel are widely used because of their portability and multifunctionality. However, they have only a small screen and it is difficult for a user to select small objects in the screen. In particular, users often make mistypes in a software keyboard application since small keys are densely arranged in the screen.

To solve this problem, there have been studies on input support by zooming. Interfaces in which a user can enlarge keys by touch or swipe operation on the screen to reduce mistypes have been proposed. In [2] and [6], a user needs to touch or swipe the screen to enlarge keys every time when the user enters a character. In [3], when a user selects a part of the keyboard by swiping the screen, the keys in the area are enlarged and the other keys disappear from the screen. Therefore, the user needs to swipe the screen when the user wants to change the area. These interfaces force a user to perform additional operation for enlarging keys. On the other hand, interfaces which display an enlarged image of the touched region using a callout have been proposed [5, 10]. However, it takes time for a user to select target objects since the area is enlarged after the screen is touched, which causes the delay of the user’s response.

There have been studies on interfaces that support selection operation without a user’s touching the screen using a hover function. The hover function is to obtain finger positions above a touch-screen using a capacitive touch panel. The interface that enables a user to select enlarged keys using the hover function has been proposed [8]. The keys around under the user’s finger are always enlarged. However, the center of enlargement moves along with the finger and the keys move in the opposite direction to the finger movement, which may cause unstable key selection. On the other hand, there have been interfaces that enable stable object selection by fixing the center of enlargement. These interfaces enlarge objects around the target position when a finger comes close to the screen [4, 12]. However, when some objects are enlarged, some other objects disappear from the screen. Therefore, these interfaces also force a user to perform additional operation to look for the next target object when it is not displayed. On the other hand, an interface that enables a user to select a target object accurately by displaying a cursor that moves along with a finger has been proposed [7]. This interface is effective to select very small objects, but it is not clear if the interface is also effective for keyboard applications.

There have been studies on input support using prediction. An interface that predicts a target object from the movement of a mouse cursor has been proposed [9]. This interface enlarges the target object to realize quick selection of densely arranged objects. There has also been a study that predicts a target position in pointing operation using the velocity of a mouse cursor [1]. However, these interfaces only support selection operation using a mouse.

A method for reducing the latency in a touch interface by predicting touch events from finger motion has been proposed [11]. In this study, prediction is used for recognizing touch input, not for input support.

In this paper, we propose PZBoard (Predictive Zooming Keyboard) which predicts a target position from the finger movement above the touch-screen of a mobile
device and that enlarges a part of keyboard around the position. A user needs not to perform additional operation for enlarging keys. We use a hover function of the mobile device to obtain the finger position above the touch-screen. While a finger is detected, a part of the keyboard around the finger position are enlarged. When the user moves his/her finger slowly, the region to be enlarged moves along with the finger with the center of enlargement fixed to the initial position. The user can select keys stably around the center of enlargement since the keys do not move even when the finger moves. When the user moves his/her finger fast to enter a distant key, the target position is predicted and the center of enlargement moves to the predicted position. Using prediction, the system can start drawing the screen before the finger reaches the target position, which reduces the system’s latency and quickens the user’s response.

The proposed interface does not force a user to perform additional operation for enlarging keys, and enables stable selection of both near and distant keys by fixing or changing the center of enlargement based on the velocity of the finger.

**Predictive Zooming Keyboard**

*Prediction*

We conducted a simple experiment to measure the velocity change of the finger movement. Figure 2 shows a part of the velocity change when a user is entering characters. Focusing on the velocity change between touches, it has a symmetric waveform. In our interface, the waveform is used to determine the timing to start prediction and to predict a target position. Figure 3 shows a model of the velocity change. Prediction is performed when the velocity is greater than \( v_{th2} \) and the acceleration becomes negative. The integral of velocity vectors from the time when the velocity becomes greater than \( v_{th1} \) to the time when prediction is started is calculated, and the predicted position is calculated by adding this value to the finger position when the prediction is started. When the finger travels about half of the distance to the target, the predicted position is determined before the finger reaches the target position as shown in Figure 4.

![Figure 2: Velocity change of the finger movement](image)

![Figure 3: Model of the velocity change](image)

![Figure 4: Calculating a predicted position](image)
When a finger is detected, the center of enlargement is set at the position under the finger and the keyboard is partly enlarged around the finger position. Since the keys in the software keyboard of a standard smartphone are vertically long and their width is shorter than that of a finger, users often mistypes horizontally adjacent keys to the target key. Therefore, the proposed interface enlarges a part of keyboard only in the X axis direction and we set the ratio to double to make the enlarged keys wider than that of the finger. The region of enlargement, in which the keyboard is enlarged, has twice the width of the enlarged keys and the height of the keyboard. When the keyboard in the region is enlarged, the keyboard in other regions is shrunk so that a user can see all keys.

Figure 5 shows the state transition diagram of our interface. When a finger comes close to the screen, the device detects the finger and a part of the keyboard around the finger position is enlarged. When the user moves his/her finger slowly, the region of enlargement moves along with the finger with the center of enlargement fixed to the initial position. The user can select keys stably around the center of enlargement since the keys do not move even when the finger moves (Figure 6a). When the user moves his/her finger fast to enter a distance key, the target position is predicted and the center of enlargement moves to the predicted position (Figure 6b). Using prediction, the system can start drawing the screen before the finger reaches the target position, which reduces the system’s latency and quickens the user’s response. When a user pulls his/her finger away from the screen and no finger is detected, the keyboard is displayed in the original size.

Since the keyboard outside the region of enlargement is sometimes too shrunk and the keys are almost invisible, we add an exceptional rule for screen edge regions. When the finger is in any of the screen edge regions, which are set near the left and right edges of the screen, for a certain period of time, the center of enlargement is gradually moved to the edge of the screen (Figure 7).
**Implementation**

The proposed interface was implemented to a Samsung Galaxy S4 smartphone. The smartphone has a 1920-by-1080 pixel touch-screen display. It can detect a finger position up to about 2 cm above the screen using the AirView feature. The operation system was Android Lollipop version 5.0.1 and the finger position in the air can be obtained using a library which is provided by the Android SDK, and the obtained finger position can be used in the application. We used the QWERTY layout software keyboard of the Google Japanese Input application and the keyboard is rendered and enlarged using OpenGL ES 2.0.

Figure 8 shows that a user was entering characters using the proposed interface. The user was able to enter target keys accurately without mistyping adjacent keys. Since the keys around the center of enlargement do not move, the user was able to select the keys stably. When the user entered a distant key, the region around the target key was enlarged with short latency and the user was able to enter the key quickly.

**Conclusion**

In this paper, we proposed PZBoard which predicts a target position from the finger movement above the touch-screen of a mobile device and that enlarges a part of the keyboard around the predicted position. The interface does not force a user to perform additional operation for enlarging keys, and enables stable selection of both near and distant keys.

In future work, we will refine the prediction algorithm to improve the accuracy, and conduct an experiment for evaluating the usability of the proposed interface.
References


