

LongPad: A TouchPad Using the Entire Area below the Keyboard of a Laptop Computer

Jiseong Gu, Seongkook Heo, Jaehyun Han, Sunjun Kim and Geehyuk Lee

Department of Computer Science

KAIST, Daejeon, South Korea

gstarcastle@gmail.com, seongkook@kaist.ac.kr, {jay.jaehyun, kuaa.net, geehyuk}@gmail.com

ABSTRACT

In this paper, we explore the possibility of a long touchpad that utilizes the entire area below the keyboard of a laptop computer. An essential prerequisite for such a touchpad is a robust palm rejection method, which we satisfy using a proximity-sensing touchpad. We developed LongPad, a proximity-sensing optical touchpad that is as wide as a laptop keyboard, and implemented a palm rejection algorithm that utilizes proximity images from LongPad. In a user study conducted, we observed that LongPad rejected palm touches almost perfectly while participants were repeating typing and pointing tasks. We also summarize the new design space enabled by LongPad and demonstrate a few of the interaction techniques it facilitates.

Author Keywords

LongPad; proximity-sensing; per-finger force sensing; palm rejection; bimanual interaction

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): User Interfaces – Input devices and strategies

INTRODUCTION

Even though new laptop computer designs are being introduced into the marketplace every day, they all tend to conform to that of the stereotypical traditional laptop computer, i.e., they are in the shape of a clamshell with an LCD display on one half and a keyboard and touchpad on the other half. In particular, the touchpad is typically located in the center, below the keyboard. It appears that the current location of the touchpad has been taken for granted as the most logical choice to avoid getting in the way of the hands while typing on the keyboard. It is in fact regrettable to see the size and position of the touchpad on a laptop being constrained by the requirement that it should not be in the way of the hands during typing activities. It is not difficult to imagine new possibilities that may actualize if the touchpad could grow beyond the constraints of typing hands. For instance, the current touchpads are too small to

accommodate two hands. Bimanual interaction techniques on a touch surface are subjects with a long history [23], and would become possible if the touchpad could grow beyond the constraints of typing hands. Intrigued by the myriad possibilities inherent in a larger touchpad, we consider an extreme case of the touchpad occupying the entire area below the keyboard. As shown in Figure 1, the touchpad under consideration has a long form factor; we therefore call it “LongPad”.



Figure 1. LongPad: a touchpad occupies the entire area below the keyboard of a laptop computer.

As mentioned above, one of the primary benefits of such a long form factor is the fact that it allows users to use both hands simultaneously and enables them to utilize their bimanual skills. Another benefit that we consider more important in practice is that laptop users will feel less constrained if the touchpad is expanded to cover the entire surface below the keyboard. The current position of the touchpad is not in fact the best for either left-handed or right-handed users. A better place for the touchpad may be below the right side of the keyboard for right-handed people, or below the left side of the keyboard for left-handed people. In fact, in a study done on the effects of the touchpad location of a laptop computer, experimental results showed that the current touchpad position is not optimal in any respect [20]. A long touchpad that covers the entire area below the keyboard would give the hands of laptop users more freedom.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27 – May 2, 2013, Paris, France.

Copyright 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

An obviously essential prerequisite for the realization of our LongPad concept is a robust “palm rejection” algorithm, i.e., users should be able to use the keyboard without worrying about the effect touching the touchpad under their hands will have. In fact, many large touchpads these days have a similar problem, which they try to resolve by using a pattern matching algorithm to filter out accidental touches that may occur while a user is typing. However, because this type of algorithm is not yet perfect, some laptop models provide an explicit physical button to disable the touchpad. After considering possible palm rejection solutions for LongPad, we finally chose to use a proximity-sensing touchpad, as we felt that touch pattern alone would not provide sufficient information for robust palm rejection. In contrast to touch pattern, a hand image from a proximity-sensing touchpad is more informative, and we believe that it would enable more reliable palm rejection. Although we do not claim that the proximity-sensing feature of a touchpad is essential for reliable palm rejection, we can show experimentally that it is highly useful for easy, robust implementation of the palm rejection algorithm that is essential for the realization of the LongPad concept.

In this paper, we describe the results of our research into the feasibility of LongPad. First, we present the design of the hardware of LongPad, a proximity-sensing and force sensing optical touchpad. In fact, an optical touchpad with both capabilities is not common, and therefore the hardware design itself may be counted as one of the technical contributions of the current research. Second, we present a palm rejection algorithm that utilizes the proximity data from LongPad, and show that the algorithm is robust by means of a user test. Third, we present the new possibilities opened up by LongPad. We first summarize the new design space enabled by the new features of LongPad, and then present the results of our prototyping study that was designed to let us experience a few representative possibilities using LongPad. Finally, we conclude this paper with a summary of the contributions made by the current research.

RELATED WORK

A touchpad is the most common pointing device for a laptop computer. Its usability has been studied and compared with that of other pointing devices. A mouse is clearly superior to other pointing devices [12], so most of the comparison studies have been done between touchpads and other embedded pointing devices, such as trackpoints and trackballs. Batra et al. [3] compared a touchpad, a trackpoint, and a trackball and showed that the touchpad and the trackball were superior to the trackpoint. More recent studies [31, 32, 33, 34] have also shown that touchpads are superior to trackpoints.

While there have been many studies done on the usability of touchpads, relatively little attention has been paid to the positioning of touchpads on laptops. Touchpads are almost always in the center and below the keyboard on a laptop,

while they are often located on the right side of the keyboard in external keyboard-touchpad combinations. Kelaher et al. [20] conducted a unique study on the effects of touchpad location on a laptop computer. The study explored the effect of the location of the touchpad (top center, top right, right side, bottom center, and bottom right) on extremity posture, discomfort, preference, and performance. The touchpad location was found to have a significant effect on each of these measures. These results infer the need for a large surface touchpad because users would then be able to select their preferred location.

A touchpad has the advantage of being able to support multi-touch gestures such as pinching and spreading. Various multi-touch sensing technologies have been introduced since the first multi-touch touchpad by Lee and others [23]. iGesturePad [36] is a large multi-touch touchpad that facilitates multi-touch gestures and two-handed text entry. Visual Touchpad [25] is an example of a large surface touchpad that uses cameras. Its creators demonstrated that a variety of one- and two-handed multi-finger gestural interaction techniques can exploit the capacity of a camera-based large touchpad. TactaPad [35] is another example of a multi-touch touchpad based on a camera. ZeroTouch [27] is an example of an optical multi-touch sensing technology using infrared emitters and receivers. Its creators used it to demonstrate multi-touch capability on a large surface with an optical sensing frame. Unlike reflection-based optical multi-touch sensing technologies, it is based on an optical occlusion principle, and therefore it requires only one-dimensional arrays of emitters and receivers around a touch-sensing area. While there are many examples of large surface, multi-touch touchpads, their application in combination with a physical keyboard on a laptop is not common.

Force sensing and proximity sensing on a touch-sensing surface are being explored to enrich the vocabulary of touch interfaces [10, 15, 17, 19, 24, 28, 29, 36]. The UnMousePad uses interpolating force-sensitive resistance to implement a multi-touch touchpad that can sense the force of each touch [29]. Hilliges et al. [15] proposed a way to implement a force-sensitive interactive surface using a latex pouch filled with black ink. SmartSkin [28] demonstrated the potential of the proximity-sensing capability of a sensitive capacitive sensing technology. ThinSight [17, 19] and FlexAura [24] use multiple sets of infrared emitters and receivers to realize a sensing surface that can detect fingers near the surface. RemoteTouch [10] uses matrices of infrared LEDs and phototransistors to facilitate hover and touch tracking. None of these technologies, however, can sense proximity and forces at the same time.

Many researchers have studied bimanual interactions on computer interfaces [8, 13, 22]. Buxton [8] conducted two experiments and discovered that performance can be improved by splitting compound tasks into simple two-handed tasks. Guiard [13] observed that most tasks are

accomplished using two hands with different roles and proposed a kinematic chain model, which is the basic principle of asymmetric bimanual interaction. Kurtenback [22] implemented a bimanual drawing application, which was shown to improve the user experience. While bimanual interaction is an old research topic, its potential benefit when applied specifically to a laptop environment in combination with a keyboard has not yet been fully explored.

As the size of touchpads increase, the possibility of unintended touches also increases. Block et al. [6] conducted a survey on the preference between a built-in pointing device—a touchpad in most cases—and an external mouse. They discovered that one of the reasons an external mouse was preferred was the accidental click triggered on the built-in pointing device while typing. Unintended touches are often due to palm contact and therefore a good palm rejection method is important. Kremin et al. [21] introduced a capacitive touchpad that rejects palm contacts based on the relative difference in signal levels. Hill [14] proposed a method that classifies touch inputs as accidental if they come from a corner, side, top, or a combination of these regions. These techniques, however, are not applicable to large surface touchpads because the wide area of a palm will frequently be placed on the large surface touchpad. Hotelling [18] placed an additional proximity sensor array between the keyboard and a wide touchpad in order to determine whether a contact was intentional or unintentional. This may be applicable to large surface touchpads, but it requires an additional sensor array and its feasibility still needs to be verified.

LONGPAD IMPLEMENTATION

In this section, we describe the implementation of LongPad, including a palm rejection method. First, we summarize some of the major technical requirements for LongPad.

Proximity sensing: In the discrimination of fingertips from other parts of the hand, a proximity image provides richer information than a contact image. The required range of proximity sensing depends on the types of discrimination tasks required. If it is required that different fingers be distinguished, it will be necessary to track not only the fingers that are in contact with the touchpad but also other fingers hovering over the touchpad. In this case, the required range of proximity sensing would be as large as a few centimeters. As the goal of our project is the implementation of a robust palm rejection method, it is required only that fingertips be distinguished from other parts of the hand. Therefore, we set the proximity-sensing range requirement to approximately 3 cm.

Touch sensing: Touch sensing is, of course, the first requirement of any touchpad. One might think that touch sensing is a kind of proximity sensing and would come at no extra cost when proximity sensing is possible. However, this is only true when the precision of the proximity sensing is very high. In practice, the output of proximity sensing is

not only dependent on the precise distance between the skin and the touchpad, but also on other variables such as skin area and the color of the skin. Therefore, it is usually difficult to tell whether a finger is in contact with the touchpad from the output of proximity sensing alone. Therefore, for robust detection of a touch event, an additional sensing mechanism is usually required.

Per-finger force sensing: A selection operation is usually done using a separate button such as the left-mouse button in the case of an ordinary touchpad. In some recent large touchpads, a button mechanism is embedded under the touchpad, and users can press the touchpad to perform selection operations. In the case of LongPad, a separate button is not desired because one of the goals of LongPad is to allow the position of the hand to be unconstrained. A button mechanism under the touchpad would also not be practical due to the size of LongPad and also because a selection operation should now be a “per-finger” operation. Therefore, we concluded that per-finger force sensing is a necessary requirement for LongPad.

Thin form factor: This is not a functional requirement but a practical constraint that we have to consider in choosing a sensing technology for LongPad. For instance, a camera may be an easy choice for realizing proximity sensing but it was excluded because LongPad would not be thin anymore if it contained a camera-based optical structure for uniform proximity sensing over its long surface.

In the following subsections, we describe the hardware design and the image processing done by LongPad. These are in fact the result of iterations of redesigns to meet the above requirements.

LongPad Hardware

Figure 2 shows simplified sketches that illustrate the operating principles of proximity and force sensing in LongPad. Figure 2a is a cross-sectional side view of the layers in LongPad, while Figure 2b is the corresponding top view. The first layer is a transparent elastic sheet (EXSEAL Hyper-Gel / Asker-C 50) with a thickness of 3 mm. The transparent elastic sheet provides a touch surface and also acts as an optical planar waveguide for force sensing. The second layer is an 8×32 matrix of infrared LEDs (with a peak wavelength of 950 nm). The LEDs are turned on and off one at a time sequentially in a row-major order. Although there are many LEDs, only one LED is on at a time, and therefore the electric power that they consume is equivalent to that of one LED. The third layer is a 9×33 array of infrared phototransistors. These phototransistors are wired in parallel and act as a single large-area photo sensor, which we denote as S1. Another set of infrared phototransistors are located around the transparent elastic sheet in order to measure light exiting from the sides of the sheet. These phototransistors are also wired in parallel and also act as a single, long photo sensor, which we denote as S2.

Proximity sensing: A near-sighted image of hands over LongPad can be obtained by sampling the output of photo sensor S1 while the infrared LEDs are turned on and off one at a time sequentially in a row-major order [10]. A sample image obtained this way is shown in Figure 3b. As shown in Figure 3a, two hands are over LongPad, with some of the fingertips more than 3 cm away from the touch surface. Each pixel value is the output of S1 for each LED, and therefore the size of the image is 8×32 . The proximity-sensing range is mainly determined by the directivity of the LEDs. In the current implementation, the half-intensity angle of the LEDs is $\pm 12^\circ$ and, in this case, it was possible to discriminate fingers from the background when the fingers were approximately 3 cm above the touch surface. The frame rate of the current prototype is about 80 fps. Considering that the current prototype uses a low-end microcontroller (PIC18F2523 with a 20 MHz clock) and a built-in ADC, doubling the frame rate is not a technical issue and can be readily done if desired.

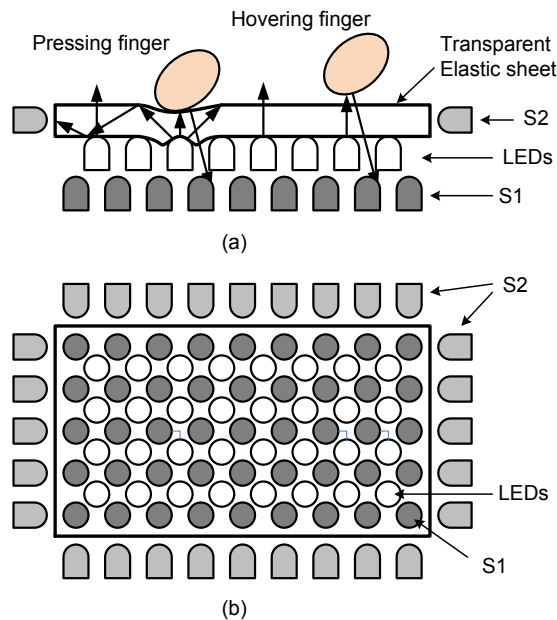


Figure 2. Structure of the hardware of LongPad: (a) cross-sectional side view, and (b) top view. White symbols represent LEDs, dark gray symbols represent back phototransistors (S1), and light gray symbols represent side phototransistors (S2).

Force sensing: The optical interface between the LEDs and the transparent elastic sheet changes as the finger force on the sheet changes. As the transparent sheet is elastic, it is deformed when a finger force is applied, as illustrated in Figure 2a. The result is the frustration of the lens structure (dome) of the LED. Prior to this happening, light from the LED is focused by the lens structure of the LED and propagates vertically. After the frustration of the lens structure, however, light from the LED cannot be focused and propagate vertically so it enters the transparent sheet (a planar waveguide). Once light from the LED enters the transparent sheet, it propagates to the edges of the sheet due

to total internal reflection and reaches photo sensor S2. The greater the force applied to the sheet, the more light enters the transparent sheet and the more light reaches S2. As in the case of proximity sensing, a force distribution image can be obtained by sampling the output of sensor array S2 while the LEDs are turned on and off one at a time sequentially. A sample force image is shown in Figure 3c. The brightness of each pixel is proportional to the force applied on each LED, and therefore the size of the image is also 8×32 px.

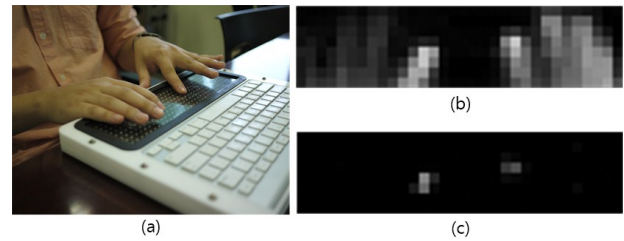


Figure 3. LongPad output images: (a) hands over LongPad, (b) proximity image from S1, and (c) force image from S2.

Touch sensing: In principle, touch sensing may be done either by proximity sensing or by force sensing. However, as we mentioned already, touch sensing by proximity sensing is not reliable because proximity sensing is not only affected by the distance to the finger but also by other parameters such as the area and the properties of the finger. Therefore, it may be more reliable to use force sensing for touch sensing. The effectiveness of this option depends of course on the sensitivity of force sensing. In our current implementation, a force of about 0.7 N is needed to produce a noticeable output from S2 that is above the noise floor, and this can be used as the threshold for touch detection. This touch sensitivity is good enough to detect selection operations. However, in a pilot study conducted, we observed that the finger force usually decreases during dragging operations and, in many cases, goes below the aforementioned touch threshold. Figure 4a is an illustration that depicts this problem and also our solution. The thick lines in the figure show the changes in the outputs of S1 (proximity sensing) and S2 (force sensing). In the figure, a finger approaches the pad at time A, and starts to touch the pad at time B. After the onset of the touch, the force of the finger on the pad increases and reaches a maximum at time C. Before time C, however, the output from S2 exceeds touch threshold, $S2_{touch}$, and a transition to the touched state occurs, as shown in the state diagram in Figure 4b. As the finger starts to slide on the pad, the output from S2 decreases and may go below the touch threshold and, therefore, the output from S2 is not a reliable signal to use to determine the transition back to the released state. We therefore decided to allow the transition to the released state when the output from S1 goes below $(S1_{touch} - \Delta)$, where $S1_{touch}$ is the value of S1 at the transition to the touched state and Δ is a small offset value. In summary, we realized touch sensing by combining proximity sensing and force sensing.

In the actual implementation of the LongPad image processing algorithm, which we will describe in the next section, we applied an interpolation operation and a convolution operation with a fingertip mask to the output of S1 before we used it for touch detection. After fingertip locations were found, we proceeded to determine the touch states of the fingertips. The basic idea of combining proximity sensing and force sensing for better touch sensing was unchanged and was shown to be effective in an experiment that we describe later. One problem with this approach, however, is that the transition back to the released state is slightly delayed compared with the actual release time. In practice, however, this did not cause a problem in either flicking or drag-and-drop operations.

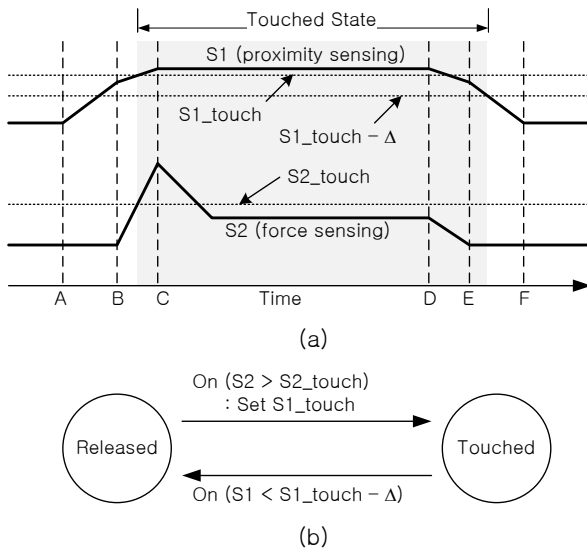


Figure 4. Touch sensing using both S1 (proximity sensing) and S2 (force sensing): (a) The changes in S1 and S2 during a dragging operation and (b) transition diagram for touch detection.

Omitting the implementation details of the LongPad hardware, we mention here a few points that are important for the reproduction of our results. First, it was necessary to use a differential sensing strategy to reduce the effect of ambient infrared light. For both S1 and S2, we measured the sensor output once with an LED on and another time with an LED off, and used the difference between the two measurements. Second, it was necessary to add a thin low-friction sheet (PVC) to the transparent elastic sheet in Figure 2 because the elastic sheet was too sticky and so was unable to facilitate dragging operations as is. The addition of the thin PVC sheet to the elastic sheet did not adversely affect the waveguide function of the elastic sheet. Third, the material chosen for the elastic layer is very important for proper force sensing. Both proper hardness and good restoration property are key factors in the successful realization of responsive force sensing.

LongPad Image Processing

Given a proximity image and a force image from LongPad, we perform a series of image processing steps to identify and localize fingers. The overall image processing steps are shown in Figure 5. First, the 8×32 proximity image is scaled up by a factor of four using a bi-cubic interpolation method. The next image processing steps branch into two flows: one to identify hands and the other flow to localize fingertips.

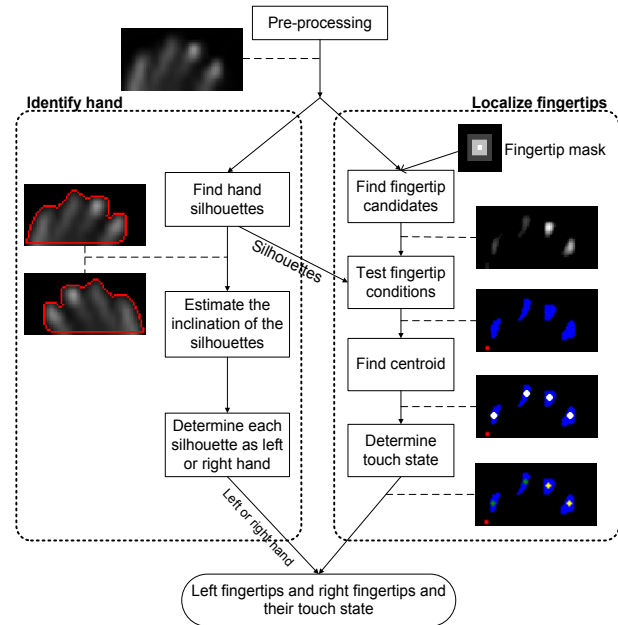


Figure 5. The overall image processing steps of LongPad

To identify hands, a low-value threshold is used to cut out a hand silhouette from the interpolated proximity image, and then its bounding box is calculated. By comparing the bottom part of the silhouette and the bottom side of the bounding box, the inclination of the hand is determined. Based on the sign of the inclination it is determined whether the silhouette is for the left hand or for the right hand. This approach is based on the observation that each hand tends to incline toward the opposite side when people use a laptop.

To localize fingers, matched filtering is applied to the interpolated proximity image. Since a finger width is approximately 4 to 6 px in the image, we made a fingertip mask, as shown in Figure 5, where a square with a 5-px width has positive values and the background has negative values. After filtering with the mask, we obtain an image that has outstanding peaks at the fingertips and relatively low values in the other parts of the hand. A high-value threshold is then applied to this image to find blobs. Each blob is assumed to be a fingertip candidate, and small blobs due to noise are discarded. Fingertip candidates that do not satisfy the following two conditions are rejected: 1) the fingertip is always near the top-side of a hand silhouette, and 2) the top n brightest pixels of each blob fit in a small square. The top n brightest pixels of each blob are then used

to determine the centroid of the blob. The touch state of each blob is determined by using both the interpolated force image and the interpolated proximity image, as explained in the previous section. The final output is the locations of fingertips, their memberships to the left or right hand, and their touch states.

LONGPAD EVALUATION

We conducted two experiments to evaluate the efficacy of LongPad's palm rejection.

Experiments

We recruited 11 university students (four females and seven males, all right-handed, mean age = 22.7 years) who had experience using touchpads on laptop computers. A 17 inch monitor with a resolution of 1366×768 px, a small keyboard, and LongPad were used to form a mockup of a laptop, as shown in Figure 6. As LongPad is capable of per-finger force sensing, a finger force over a certain threshold was used as a selection action. This force threshold for the selection action, as well as other parameters such as control-display ratio, was determined based on the results of an earlier pilot test with three participants

In the first experiment, participants alternately repeated a typing task and a pointing task, as depicted in Figure 6. In the typing task (Figure 6a), participants were asked to transcribe a sentence shown on the screen using the keyboard. In the pointing task (Figure 6b), participants were asked to point and select a blue circle and a red circle in sequence. We prompted participants to alternate their hands in this task by showing a hand symbol on the left or right side of the screen. The repetition of the two tasks led participants to experience various hand shapes and postures. Each participant was asked to finish six blocks, where each block consisted of 50 repetitions of the two tasks. It took between 8 and 12 minutes to finish a block, and we provided a two-minute break between blocks. The text to transcribe in the typing task and the location of the target in the pointing task were randomly chosen.

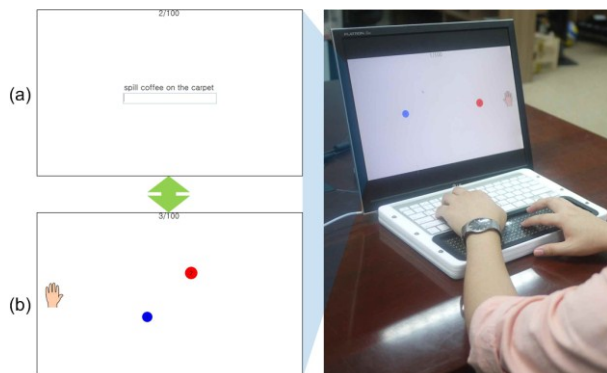


Figure 6. Screenshot of the first experiment: Screen while performing (a) the typing task and (b) the pointing task.

After the first experiment, participants took a two-minute break and then were given a brief introduction to the second experiment. In the second experiment, we asked

participants to buy an AAA rechargeable battery from an online store. The task involved entering information, such as a physical address, a name, and an e-mail address, and manipulating GUI widgets such as buttons, check boxes, and a list box. Left-hand pointing was mapped to vertical scrolling, while right-hand pointing and pressing were mapped to manipulation of the cursor and selection of a target, respectively. We provided the participants with all the information required to do the shopping.

During the test, we logged both proximity images and force images and image processing results. We also videotaped the movement of the hands using the prototype, and the computer screens shown to participants. When all the tasks were finished, we asked them to answer a brief questionnaire about their experience using LongPad.

Results

Figure 7 shows the error rates from the first experiment. An error rate here represents the number of typing trials in which unintentional touch occurred divided by the total number of typing trials. Except for participant T1, the error rate was less than 1% for all participants. The average over all participants was 0.42%.

There were three sources of error. The first was unintentional thumb touch while moving the cursor with the arrow keys. This occurred eight times in total. The second was unintentional thumb touch while moving a hand toward a keyboard, which occurred a total of four times. The third was due to the sleeve of one participant being wrongfully recognized as a fingertip. This occurred three times for the participant. In fact, the thumb touches in the first and second error scenarios had to be handled as valid touch inputs because people sometimes use a thumb to point while resting other fingers on the keyboard. If the thumb touches had been counted as valid touch inputs, the error rate would have been 0.09%.

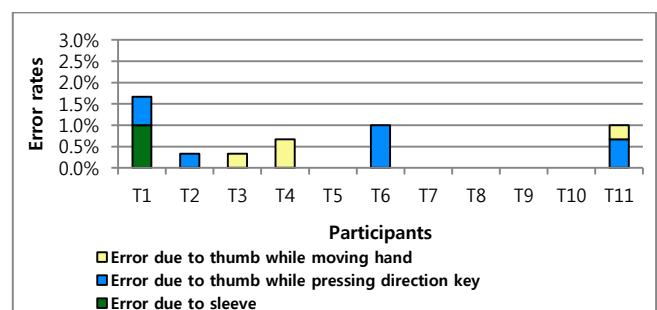


Figure 7. Error rates of LongPad's palm rejection algorithm in the first experiment

We made some interesting observations during the experiments. The first observation was that a participant who had long fingernails had difficulty while doing clutching on LongPad. Touching a touchpad with a fingernail is not usually detected as a touch on a capacitive touchpad. In the case of LongPad, however, touching with a fingernail is almost as effective as touching with a fingertip.

Therefore, the participant who had long fingernails experienced unintended cursor movement back to the initial location while clutching. The second observation was that participants tended to touch the surface with additional fingers unintentionally when selecting a target with a finger force. Approximately 4.67% of the pressing operations were performed with additional fingers touching the surface. This may be a more natural hand skill considering the structure of the hand, but could be observed only in this case because selection by pressing was possible due to the per-finger force sensing of LongPad.

We also received positive and negative comments during the interviews. Some participants commented that they usually do not rest their palm on the laptop when they use a laptop keyboard in order to avoid interference with the touchpad. They said it was very comfortable to rest their palms on LongPad as they can when they use a desktop keyboard. Some participants said it was good that they did not need to do clutching when moving a cursor horizontally. Others said that it was good that they could rest one hand while moving the cursor with the other hand, and some participants said it was tiring to select a target with a pressing operation when they needed to do it repeatedly. One participant said he would eventually use the middle part of LongPad even though LongPad enables him to use the entire area below the keyboard as a touch interaction surface.

LONGPAD INTERACTION

New Interaction Possibilities

As LongPad has been shown to be unaffected by typing hands, the entire space below the keyboard can now be considered to be available for new interaction possibilities. The features of LongPad and new interaction possibilities enabled by the features are summarized in Table 1.

First of all, as the name implies, the width of LongPad is matched to the width of the screen. The new form factor can enable new possibilities that were not possible with ordinary touchpads. First, touchpad operations are less constrained in position. Right-handed people can use the right side of LongPad while left-handed people can use the left side. Second, one-to-one mapping to a long GUI control is possible. For instance, a horizontal edge of LongPad may be mapped to the horizontal taskbar of MS Windows, enabling quick task switching. Other examples of long controls are the timeline control of a movie player and the main menu bar of full-screen applications. Third, the wide area of LongPad enables rich bimanual interaction techniques. Bimanual interaction research has a long history [5, 6, 7], and there are numerous bimanual interaction technique examples. Drawing with a dominant hand while moving the canvas with a non-dominant hand is a typical example [37].

In addition to the change in form factor, LongPad has additional features—namely, proximity sensing and per-

finger force sensing—that are not common features of ordinary touchpads. These features are not the focus of the current paper but are the technical requirements of LongPad, e.g., for robust palm rejection. Nevertheless, these are very useful features that can greatly expand the interaction possibilities of LongPad, so we also list new possibilities facilitated by these features in Table 1 and demonstrate some of them in our prototyping study.

LongPad features	New interaction possibilities	Interaction technique examples
Long form factor	Less constrained operations	Right-handed people can use the right side of LongPad.
	One-to-one mapping to the width of the screen	Quick task switch, quick menu access, video player timeline control.
	Bimanual interaction	Drawing with a tool in the other hand [5, 6, 7], manipulation while scrolling [37]
Proximity sensing	2 + 1/2 dimensional interaction	Over-the-surface 3D object manipulation [16], continuous space interaction [26]
	Preview and guides	Gesture guides and just-in-time widgets [1]
	Ten finger identification	Per-finger hotkeys [30], finger-aware drawing [4]
Per-finger force sensing	Normal force operations, multi-point force operations	Force-sensitive virtual piano [29], using force as depth dimension [29], layering using force [11]

Table 1. Features of LongPad and new interaction possibilities facilitated by the features.

LongPad Interaction Technique Examples

Among the interaction technique examples summarized in Table 1 for LongPad, we selected a few examples that can be implemented faithfully using the current LongPad prototype. We did not choose a 2 + 1/2 dimensional interaction technique or a ten finger identification scenario due to the limited proximity sensing range of the current LongPad prototype. We actually attempted to implement one such scenario but the implemented application worked only when we maintained our hands quite close to the LongPad surface, and was not appropriate for use in a user study. The finally chosen examples were 1) quick application switching, 2) video player timeline control, 3) manipulating while scrolling, and 4) force-sensitive virtual piano, and the corresponding applications are shown in Figure 8. Our goal of implementing example applications was to collect user feedback about their usefulness and usability. The first three example applications were implemented in C# on the .Net framework 4.5, and the last one was implemented in Processing 1.5.

Quick application switching

As the horizontal extent of LongPad matches well to the horizontal extent of a display, it is possible to use one-to-one mapping from the lower side of LongPad to the task bar of MS Windows. In our prototype application, the approach of a finger to the lower side from the outside of the sensing range triggered application switching mode. In this mode, a finger movement in the horizontal axis can be used to

preview or select an application window on a task bar. This allows instant access to an application using a finger hover action followed by a finger press action.

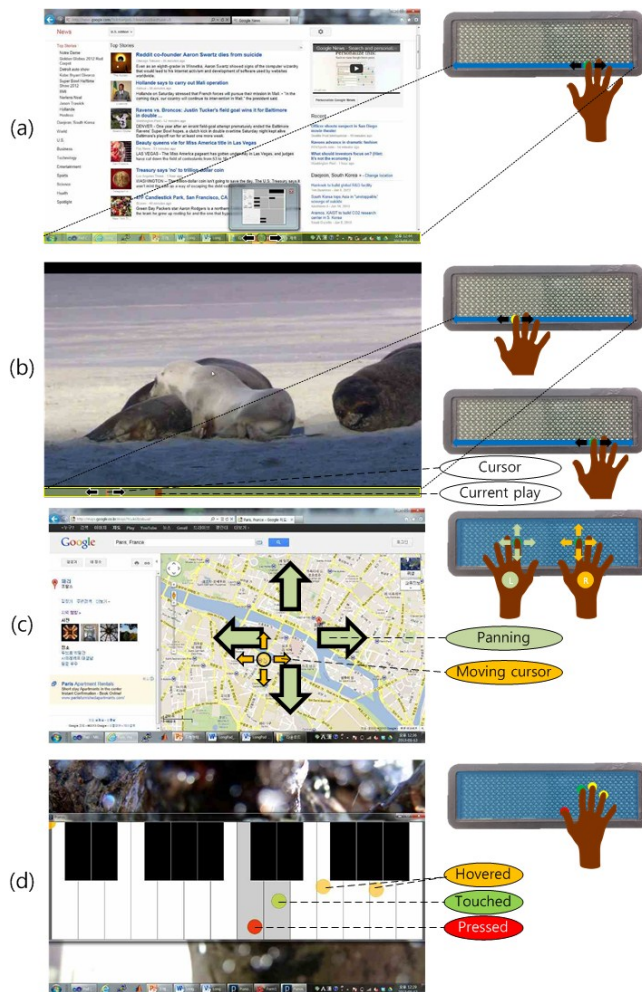


Figure 8. Four interaction technique examples with LongPad: (a) quick application switching, (b) video player timeline control, (c) manipulating while scrolling, and (d) force-sensitive virtual piano.

Video player timeline control

The long form factor provides a more continuous input on the horizontal axis. Similar to the application switching application, we can map the horizontal position of a finger touch in the lower side of the prototype to the timeline control of a video player that is in the full-screen mode. In the application, the hovering motion of a finger is mapped to move a slider and a touch of the finger makes the video jump to a desired location, thereby enabling direct access to certain positions in a video.

Manipulating while scrolling

Bimanual operations are natural in the real world and they are also common on large multi-touch devices such as multi-touch tabletop computers. Many laptop touchpads have multi-touch sensing capabilities, but only single-handed multi-touch gestures are in use [2]. In our

implementation, the prototype is wide enough for both hands to be placed on it, and hand identification is also possible. We separated the operations of the two hands. Because the dominant hand can more precisely perform manipulations than the non-dominant hand, the movement of the dominant hand was used to control a cursor and dragging using the non-dominant was mapped to scrolling.

Force-sensitive virtual piano

To manipulate multiple force-sensitive virtual keys or buttons, the detection of forces for independent touches is required. As the LongPad prototype is capable of per-finger force estimation, it is possible to support an application with multiple force-sensitive virtual key. As an example, we implemented a piano application in which the positions of fingers on or over the virtual keyboard are visualized as circular cursors. Users may then play notes under their fingers by applying a force to the corresponding finger.

Informal User Study

We conducted an informal user study to determine usability issues and to discover insights through feedback. Four participants (one female and three males, all right-handed, average age = 22.5 years) were recruited from the university. In the study, participants used the aforementioned four applications after being given brief instructions. After the study, we asked participants to answer usability questions and to choose the application they preferred the most. We also had a short interview after the experiment about how they felt about the interface.

Three participants preferred the quick application switching interaction technique. When asked their reason for choosing quick application switching, the participants answered that the dedicated lower area for application switching is easy to use and seems to be useful.

Three participants answered that they had difficulties using the piano application because it was not easy to manipulate in the vertical direction, i.e., to move between the white and black keys. Also, the vertical size of LongPad was too small to align all five fingers horizontally. We can surmise the reason for the difficulties from the coordinate mapping. The three applications—quick application switching, video player timeline control, and force-sensitive virtual piano—use absolute mapping, i.e., one-to-one mapping from the surface of LongPad to the entire screen area. Because LongPad uses the palm rest area of a laptop computer, it has a wider aspect ratio than the laptop display and the control-display ratio is lower in the vertical direction. This did not cause a problem in the first two applications because the two applications use only movement in the horizontal direction. On the other hand, the piano application uses two-dimensional movement, and therefore, the users had to be careful when making vertical movements. This problem did not matter in the case of the bimanual interaction application although it also uses a two-dimensional input space. The difference was due to the fact that the bimanual manipulation application uses relative mapping.

From the user study, we obtained the insight that mapping the horizontal input space to the display space can be a useful way to give instant access to items that are arranged horizontally. However, it is not a good idea to directly map the interaction space of LongPad to the display space. For applications requiring vertical movements, it is recommended that either the height of the display space that is mapped to the input space be narrowed or relative mapping be used.

DISCUSSION

The goal of the current study was to show the feasibility of a touchpad extending the wide space below the keyboard of laptops. Although the spatial resolution of the current prototype was sufficient for this goal, i.e., for implementing a required palm rejection algorithm and also for implementing example applications, it was not good enough for fine control of a cursor on the screen. The spatial resolution of LongPad is determined by the distance between the adjacent LEDs, and the distance, i.e., the LED interval in the current prototype is about 8mm. This interval was determined considering the expected spatial wavelength of a hand image and the Nyquist sampling requirement. We now think that we overestimated the expected spatial wavelength of a hand image and determined an LED interval to be too large to enable a satisfactory spatial resolution. This low spatial resolution seems to cause some irregularity in cursor movement especially when absolute mapping was used as in the case of the timeline control application. From this experience, we plan to use denser LED matrices with 5 mm intervals in the next LongPad prototype.

LongPad is based on optical sensing and therefore is affected by ambient light. In order to handle the ambient light issue, we used a differential sampling method as explained in the Implementation section. The differential sampling method was good enough to allow LongPad to work properly in indoor environments, both under fluorescent lamps at nights and by a window during the day. However, the current prototype does not work outdoors directly under the sun because the direct sunlight saturates the optical sensors. If it is required to handle such a case, adding a narrow-band infrared filter over the photo sensors may be helpful.

The proximity sensing range of the current LongPad prototype is about 3cm. This was sufficient for the goal of this paper but is too short to enable the full potential of the proximity-sensing feature of LongPad. For example, the proximity-sensing feature of LongPad can also be useful in supporting area-based gestures [9] and recognizing objects that are above the surface. The proximity-sensing range is mainly determined by the directivity of the LEDs, and we plan to use LEDs with the half-intensity angle of $\pm 3^\circ$ in the next LongPad prototype.

CONCLUSION

The goal of the research was to explore the possibility of a long touchpad that utilizes the entire area below the keyboard on a laptop computer. An essential prerequisite was the development of a robust palm rejection algorithm. We approached this problem by employing a proximity-sensing touchpad and showed empirically that the approach was successful. Next, we explored the new design space enabled by LongPad and demonstrated a few useful interaction techniques.

The main contribution of this paper is that this is the first research to show the feasibility of a touchpad that utilizes the entire area below the keyboard. Another contribution is that it shows the new possibilities opened up by a touchpad with such a long form factor and additional capabilities such as proximity-sensing and per-finger force sensing. Lastly, the LongPad hardware is also a contribution of this paper as a unique design example that achieves proximity-sensing, per-finger force sensing, and robust touch sensing using a single optical sensing structure.

ACKNOWLEDGMENTS

This work was supported by the IT R&D program of MKE/KEIT. [10039161, Development of core technologies for high-physicality control interfaces and personalized intelligent user interfaces based on Smart TV user experience]

REFERENCES

1. Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice G., Medusa: a Proximity-Aware Multi-Touch tabletop. In *Proc. UIST'11*, 337-346.
2. Apple, Magic TrackPad. <http://www.apple.com/magictackpad/>
3. Batra, S., Dykstra, D., Hsu, P., Radle, K., and Wiedenbeck, S., Pointing Device Performance for Laptop Computers. In *Proc. Human Factors and Ergonomics Society Annual Meeting Oct. 1998*, 42, 536-540.
4. Benko, H., Saponas, S., Morris, D., and Tan, D., Enhancing Input on and above the Interactive Surface with Muscle Sensing. In *Proc. ITS '09*, 93-100.
5. Bier, E., Stone, M., Pier, K., Buxton, W., and DeRose, T., Toolglass and Magic Lenses: the See-Through Interface. In *proc. SIGGRAPH '93*, 73-80.
6. Block, F. and Gellersen, H., Two-Handed Input in a Standard Configuration of Notebook with External Mouse. In *proc. NordiCHI '10*, 62-71.
7. Butler, C. and Amant, R., HabilisDraw DT: a Bimanual Tool-Based Direct Manipulation Drawing Environment. In *proc. CHI '04 Ext. Abstracts*, 1301-1304.
8. Buxton, W. and Myers, B., A Study in Two-Handed Input, In *proc. CHI '86*, 321- 326.

9. Choi, S., Gu, J., Han, J., and Lee, G., Area Gestures for a Laptop Computer Enabled by a Hover-Tracking Touchpad. In *Proc. APCHI'12*, 119-124.
10. Choi, S., Han, J., Lee, G., Lee, N., and Lee, W., RemoteTouch: Touch-Screen-like Interaction in the TV Viewing Environment. In *proc. CHI '11*, 393-402.
11. Davidson, P. L. and Han, J. Y., Extending 2D Object Arrangement with Pressure-Sensitive Layering Cues, in *proc. UIST '08*, 87-90.
12. Epps, B. W., Comparison of Six Cursor Control Devices Based on Fitts' Law Models. In *Proc. Human Factors and Ergonomics Society Annual Meeting Sept. 1986*, 30, 327-331.
13. Guiard, Y., Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, 1987, 19, 486-517.
14. Hill, J., US Patent US 2011/0069021, 06.14.2010
15. Hilliges, O., Kim, D., and Izadi, S., Creating Malleable Interactive Surfaces Using Liquid Displacement Sensing. In *proc. TABLETOP 2008*, 157-160.
16. Hilliges, O., Izadi, S., Wilson, A. D., Hodges, S., Garcia-Mendoza, A., and Butz, A., Interactions in the Air: Adding Further Depth to Interactive Tabletops. In *proc. UIST '09*, 139-148.
17. Hodges, S., Izadi, S., Butler, A., Rustemi, A., and Buxton, W., ThinSight: Versatile Multi-Touch Sensing for Thin Form-Factor Displays. In *proc. UIST '07*, 259-268.
18. Hotelling, S., US Patent US 7561146 B1, 08.25.2004
19. Izadi, S., Hodges, S., Butler, A., West, D., Rustemi, A., Molloy, M., and Buxton, W., ThinSight: a Thin Form-Factor Interactive Surface Technology. *Communications of ACM* Dec. 2009, 52(12), 90-98.
20. Kelaher, D., Nay, T., Lawrence, B., Lamar, S., and Sommerich, C., An Investigation of the Effects of Touchpad Location within a Notebook Computer, *Applied Ergonomics* 2001, 32, 101-110.
21. Kremin, V., Ogirko, R., Pirogov, R., Ryshtun, A., and Vallis, D., US Patent US 2012/0105361 A1, 08.19.2011
22. Kurtenback, G., Fitzmaurice, G., Baudel, T., and Buxton, B., The Design of a GUI Paradigm Based on Tablets, Two-Hands, and Transparency. In *proc. CHI '97*, 35-42.
23. Lee, S., Buxton, W., and Smith, K., A Multi-Touch Three Dimensional Touch-Sensitive Tablet. In *proc. CHI '85*, 21-25.
24. Liu, S. and Guimbretière, F., FlexAura: a Flexible Near-Surface Range Sensor. In *proc. UIST '12*, 327-330.
25. Malik, S. and Laszlo, J., Visual Touchpad: A Two-Handed Gestural Input Device. In *proc. ICMI '04*, 289-296.
26. Marquardt, N., Jota, R., Greenberg, S., and Jorge, J., The Continuous Interaction Space: Interaction Techniques Unifying Touch and Gesture on and above a Digital Surface. In *proc. INTERACT 2011*, 461-476.
27. Moeller, J. and Kerne, A., ZeroTouch: an Optical Multi-Touch and free-air interaction architecture. In *proc. CHI '12*, 2165-2174.
28. Rekimoto, J., SmartSkin: an Infrastructure for Freehand Manipulation on Interactive Surfaces. In *proc. CHI '02*, 113-120.
29. Rosenberg, I. and Perlin, K., The UnMousePad: an Interpolating Multi-Touch Force-Sensing Input Pad. In *proc. SIGGRAPH '09*, 65.
30. Sugiura, A. and Koseki, Y., A User Interface Using Fingerprint Recognition: Holding Commands and Data Objects on Fingers. In *proc. UIST '98*, 71-79.
31. Sutter, C. and M. Ziefle., How to Handle Notebook Input Devices: An Insight in Button Use Strategy. *Contemporary Ergonomics* 2003, 241-246.
32. Sutter, C. and M. Ziefle., Interacting with Notebook Input Devices: An Analysis of Motor Performance and Users' Expertise, *Human Factors: The Journal of the Human Factors and Ergonomics Society Spring 2005*, 47, 169-187.
33. Sutter, C., and M. Ziefle., Psychomotor Efficiency in Users of Notebook Input Devices: Confirmation and Restrictions of Fitts' Law as an Evaluative Tool for User-friendly Design. In *proc. Human Factors and Ergonomics Society Annual Meeting, Sep. 2004*, 48, 773-777.
34. Sutter, C., M. Ziefle, H. Luczak, and K. J. Zink. Health Hazard from Input Devices: The Diagnostics of Muscular Load and Motor Performance Revisited. *Human Factors in Organizational Design and Management* 2003, 525-530.
35. TactaPad, <http://www.tactiva.com/tactapad.html>
36. Westerman, W. Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-Touch Surface. PhD thesis *University of Delaware* 1999.
37. Yee, K., Two-handed Interaction on a Tablet Display. In *proc. CHI '04 Ext. Abstracts*, 1493-1496.