Press-n-Paste: Copy-and-Paste Operations with Pressure-sensitive Caret Navigation for Miniaturized Surface in Mobile Augmented Reality

LIK-HANG LEE, KAIST, Republic of Korea and The University of Oulu, Finland
YIMING ZHU, University of Science and Technology of China, China and Hong Kong University of Science and Technology, Hong Kong SAR
YUI-PAN YAU, Hong Kong University of Science and Technology, Hong Kong SAR
PAN HUI, Hong Kong University of Science and Technology, Hong Kong SAR and University of Helsinki, Finland
SUSANNA PIRTTIKANGAS, The University of Oulu, Finland

Copy-and-paste operations are the most popular features on computing devices such as desktop computers, smartphones and tablets. However, the copy-and-paste operations are not sufficiently addressed on the Augmented Reality (AR) smartglasses designated for real-time interaction with texts in physical environments. This paper proposes two system solutions, namely Granularity Scrolling (GS) and Two Ends (TE), for the copy-and-paste operations on AR smartglasses. By leveraging a thumb-size button on a touch-sensitive and pressure-sensitive surface, both the multi-step solutions can capture the target texts through indirect manipulation and subsequently enables the copy-and-paste operations. Based on the system solutions, we implemented an experimental prototype named Press-n-Paste (PnP). After the eight-session evaluation capturing 1,296 copy-and-paste operations, 18 participants with GS and TE achieve the peak performance of 17,574 ms and 13,951 ms per copy-and-paste operation, with 93.21% and 98.15% accuracy rates respectively, which are as good as the commercial solutions using direct manipulation on touchscreen devices. The user footprints also show that PnP has a distinctive feature of miniaturized interaction area within 12.65 mm * 14.48 mm. PnP not only proves the feasibility of copy-and-paste operations with the flexibility of various granularities on AR smartglasses, but also gives significant implications to the design space of pressure widgets as well as the input design on smart wearables.

Additional Key Words and Phrases: Augmented Reality, Human-Computer Interaction, Target Acquisition

1 INTRODUCTION

Copy-and-Paste is one of the standard yet most frequently used features in the modern computing devices [54], which is designed for replicating a piece of contents, in the forms of texts or graphics, from one location to another in digital interfaces. Copy-and-Paste operation is originated from the desktop metaphor with the tangible interfaces of keyboard and mouse duo, and further migrates to the touchscreen interfaces on mobile devices such as smartphones and tablets. As copy-and-paste operation involves multiple steps, we see the ease of interaction is at the trend of escalating difficulties, as follows. The desktop metaphor can accommodate various steps by providing different buttons, and achieves direct manipulation on the text contents with mouse pointers. However, the operations on the mobile devices with limit-size touchscreens become more complicated, in which a variety of tap gestures are employed to distinguish various operations with the targeted texts, e.g. double-tap or tap-and-on. When the operations come to the wearable headset computers such as augmented reality (AR) smartglasses, the touch interfaces diminish and even unavailable.

Authors' addresses: Lik-Hang Lee, KAIST, Republic of Korea and The University of Oulu, Finland; Yiming Zhu, University of Science and Technology of China, China and Hong Kong University of Science and Technology, Hong Kong SAR; Yui-Pan Yau, Hong Kong University of Science and Technology, Hong Kong SAR; Pan Hui, Hong Kong University of Science and Technology, Hong Kong SAR and University of Helsinki, Finland; Susanna Pirttikangas, The University of Oulu, Finland.

Therefore, alternative modalities and solutions, serving as a replacement of the touchscreen, are widely adapted for the user interaction on augmented reality smartglasses [28]. Instant extraction of contents from the physical world offers convenience for further editings, such as converting physical objects into images[1], supported by BaseNet [37]. In fact, the interaction design of AR smartglasses is still in its infancy, where the design space of copy-and-paste operation on AR smartglasses has not yet been explored. And specifically for text, the commercial solutions are not yet ready on the AR smartglasses in mobile situations, and the users unwillingly work with the sedentary keyboard and mouse duo[2]. For example, the users with AR smartglasses have no easy way to perform copy-and-paste on web browsers[3].

In this paper, we propose two system solutions, namely Granularity Scrolling (GS) and Two Ends (TE), and implement an experimental prototype on AR smartglasses, named Press-n-Paste (PnP). PnP is a multi-modal interaction solution leveraging touch-sensitive and pressure-sensitive surface, and these two modals work complementary within a thumb-size button, as follows. The touch-sensitive surface enables tap, pan and swipe gestures for confirmation and selection, while the pressure-sensitive surface enables the caret navigation and granularity management in the texts. As the direct manipulation on the see-thru displays of AR smartglasses is difficult, the pressure becomes the backbone in the two system solutions as the primary modality for indirect manipulations facilitating the caret navigation and granularity management. Users can employ the thumb press and subtle pan gestures to direct the caret position and select the text granularity in in-text environments.

In PnP, GS and TE have different ways to target the text being copied. GS consists of three non-identical steps, including the caret navigation, granularity management, and scrolling gestures before the final stage of copy-and-paste operations. In the caret navigation, an in-text caret selects the target characters. Then, granularity management enables users to choose the targeting granularity such as character, word, sentence, paragraph, as well as the entire text. Finally, the user performs scrolling gestures to capture the necessary text units at the chosen granularity. TE has two identical caret navigation steps to enclose the target text by selecting the two characters at both ends. After selecting the target text by either GS or TE, the user can do a down-swipe for the copy operation. After confirming the paste location using the caret navigation, an up-swipe accomplishes the paste operation.

The paper serves as the first effort to propose a solution for transferring texts from one location to another, with the mean of indirect manipulation on AR smartglasses. The contributions of this paper are fourfold, as follows. First, we present two novel copy-and-paste solutions named Granularity Scrolling (GS) and Two Ends (TE) for the copy-and-paste operations on AR smartglasses, where texts are the most common contents in our daily routine. Second, the investigated areas of user footprint in GS (9.96 mm * 10.51 mm) and TE (12.65 mm * 14.48 mm) prove that the copy-and-paste operations can be done on either a smartphone, which is a daily object with high user acceptability, or even smaller smart wearables (e.g. the spectacle frame of AR smartglasses and smart rings). Third, our solutions are not only practical for the sake of mobility and portability, but also demonstrate the design space of pressure widgets for complicated user interactions including copy-and-paste operations. Finally, after the 8-session training, 18 participants with GS and TE achieve peak completion times of 17,574 ms and 13,951 ms per CP operation, with accuracy rates of 93.21% and 98.15% respectively. The performance is comparably efficient as the commercial

[1] Cut and Paste your surrondings: https://github.com/cyrildiagne/ar-cutpaste

[2] Is there a voice command for hololens copy and paste?: https://forums.hololens.com/discussion/6482/is-there-a-voice-command-for-hololens-copy-paste

[3] How do you copy/paste text while browsing?: https://forums.hololens.com/discussion/1890/how-do-you-copy-paste-text-while-browsing

solutions on touchscreen devices. More importantly, TE outweighs GS that is modified from a state-of-the-art solution designated for direct manipulation, which serves as an (counter-)evidence to an alternative strategy under the setting of indirect manipulation on AR smartglasses. Also, Press-n-Paste (PnP) receives positive feedback from the participants, in terms of the usefulness, ease-of-use and intention of use. The rest of this paper is organized as follows. After a quick review of the key related work in Section 2, we describe the two solutions and implementations in Section 3. We then validate our copy-and-paste solutions through two evaluations in Section 4 and 5. We finally discuss our findings in Section 6.

2 RELATED WORK

Copy and Paste Operations have long been recognized as a challenging problem in the context of mobile human-computer interaction. In this section, we first discuss two major components, i.e. text selection and text relocation. And the challenges of designing such operations on mobile AR is as follow.

2.1 Text Selection on touch-based interfaces

The majority of the studies on text manipulation focuses on the touch-based interfaces appearing in various computing devices [1, 22]. The following examples illustrate that text manipulations on laptop computers and smartphones are never an easy task in the domain of human-computer interaction, although a laptop computer owns a spacious touchpad for controlling the caret position within in-text environments. In Push-Edge and Slide Edge [30], the tedious operations involving multi-finger gestures for scrolling and selecting texts on the touchpad of laptops are addressed. The users with an edge-based solution achieve faster in-text selection with lesser physical workloads. The interaction on a touchpad of a laptop is an example of indirect manipulation because any touchpoints on the touchpad relatively map to the computer screen. In contrast, users with touchscreen interfaces on mobile devices such as smartphones and tablets can directly touch on the target texts, which refers to direct manipulation [36]. Alternatively, other text selection methods on touchscreens rely on semaphoric gestures [9].

Although the texts on touchscreens can be accurately manipulated by the user's tapping and rubbing gestures [35], the densely packed yet small characters in the text pose challenges of user interaction and hence performance issues. For instance, the precise pointing on a specific character, which is the smallest granularity of the text, is more challenging than the small and large icons as well as menus and windows. Caret movement is difficult with touch input because of finger occlusion [47] and the imprecision of interacting with a small display using fingers [21]. One of the solutions is granularity management [10, 32] that alters the selection of textual units (e.g. character, word, sentence). Word Snapping [32] considers the character-level selection is unnecessary, and therefore the default granularity is word-level selection. The pointing on any characters of a word refers to the selection of the word. As a result, the alternation from character-level to word-level improves the selection speed and alleviates the user workloads. A most recent work proposes a text gauge [10] for text selection, namely ForceSelect, which achieves a full coverage of textual granularity in terms of characters, words, sentences, paragraphs and entire texts. Users with the text gauge employ 3D touch on an iPhone to select the suitable text granularity. The gauge provides the flexibility of reaching the targeting granularity and hence speeds up the user performance of in-text selection on touchscreens. In comparison, our proposed system solution (PnP) focuses on the whole copy-and-paste operations. We acknowledge that one of our proposed solutions, namely *Granularity Scrolling* (GS), also employs a similar pressure-sensitive text gauge to select the textual granularity. However, ForceSelect allows users to directly tap on the touchscreen (direct manipulation) to complete the text selection, a *sub-set operation of copy-and-paste*, while GS involves pointing and scrolling gestures to compensate the disadvantages from indirect manipulations on AR smartglasses, as the content on AR smartglasses does not allows direct manipulation. More importantly, our alternative solution named *Two Ends* (TE) achieves faster solutions than GS in the setting of indirect manipulation, which serves as an (counter-)evidence for a more appropriate text selection strategy with indirect manipulation.

Even though interaction design on touchscreens usually plays a crucial role in direct manipulation, our work leverages touchscreens, the most ubiquitous device nowadays, as an accompany device for indirect manipulation on AR smartglasses. Accordingly, only a miniaturized surface is taken on the touchscreen for user interactions (e.g. caret navigation) within in-text environments. In such indirect condition, we conduct a thorough investigation on two proposed copy-and-paste strategies.

2.2 Text Relocation on touch-based interfaces

Apart from the above approaches for text selection with flexible granularities, the next indispensable step is the relocation or transfer of the selected contents from one place to another [15]. Various techniques are designed to facilitate the content relocation on the 2D interfaces with their own goals. In the most fundamental interfaces on laptop computers [5], the relocation of selected texts between multiple windows is clumsy when one is overlapping with others. If the relocation target appears at the window being overlapped, the direct drag-and-drop of the selected texts from one window to another window will take more steps. To tackle the interaction barrier, the widows on the top will roll up to reveal an overlapped one.

Another example, named Citrine [46], is a text parser with the capability of finding the structure in copied text, where users can paste the structured information, which might have many segments, in a single paste operation. Furthermore, one example of the copy-and-paste operations through indirect manipulation is BezelCopy [6] that utilizes bezel gestures to select the desired sentences and perform paste operations accordingly. However, the indirect manipulations are primarily designed for people with motor disability. In a system designated for disabled people [20], named Gaze Writing, gaze pointing is applied to select the target texts and choose the copy and paste buttons in a distal screen.

Beyond the sole 2D interfaces on a single device, the copy-and-paste operations have been considered on table-size tablets for multi-user collaborations in shared surface [44]. The key issue on shared surfaces is that copy-and-paste actions of users are interleaved and confusing, because users interact simultaneously through the same surface. To implement the familiar copy-and-paste semantics on shared surfaces, users leverage their smartphones to act as their own clipboards. The shared tabletop surface is able to distinguish input from different users and their devices to resolve individual copy-and-paste operations unambiguously. Instead of a shared tabletop, Memory Stones [17] enables copy-and-paste operations between multiple touchscreen devices. In comparison, on top of the proposed text selection strategy, this paper also fills the gap in the copy-and-paste operations on AR smartglasses, in which the touchscreens on smartphones serve as a medium for the indirect manipulation of textual relocation.

2.3 Interacting with texts in augmented reality

The landscape of interaction design has been drastically changed on AR smartglasses. That is, neither the keyboard and mouse duo nor touchscreens are available in the AR smartglasses of limited form size [28]. With this disadvantage, the user interaction shifts to indirect manipulation, where the user manipulation on a sensory device will align with the visualization of the digital contents and interaction status on the head-worn display [24]. Without the appropriate interaction techniques for user interaction, the usability and hence the user acceptance to AR smartglasses would be very dissatisfied [25]. The commercial solutions of handheld controllers and trackpoints have either mobility or usability issues: the handheld controllers occupies the user's hands and hence scarify the user mobility [27, 49, 53], while trackpoints on the spectacle frame of AR smartglasses (e.g. Mad Gaze[4]) is ineffective [26]. Therefore, an emerging number of interaction techniques have been designed for AR smartglasses in recent years. Apart from commercial solutions such as speech inputs, three types of interaction modalities, including addendum sensors, mid-air gestures, and touchscreen interactions, have been commonly considered for text entry. However, eye-tracking sensors are not commonly available in AR smartglasses [25]. Gaze-based interaction suffers from unintended selections [34], and thus serves as an auxiliary modality with touch-based inputs. For example, Gaze'N'Touch [42] leverages touch-based interaction to mitigate the unintended selection for gaze pointing to the texts. Although their results are not statistically significant, their users prefer the gaze-based interaction with larger text size.

Addendum sensors are frequently employed in the form factor of gloves, which is characterized by subtle interaction [49] and easy-to-carry [27]. Digitouch [49] is a text entry system supported by two-handed gloves. A number of capacitive plates locate at the user's fingers, and thumb touches on these plates accomplish the text entry task. Similarly, another one-handed glove [27] enables users to input 26 alphabets through the thumb-to-finger interaction within the same hand. Moreover, hand gestures in mid-air feature with intuitive and natural interactions [25]. However, as the mid-air pointing suffers from coarse positioning [26] and Midas problem [18], the QWERTY keyboard becomes obsolete. Alternatively, a miniaturized 1-line interface, namely HIBEY [28], not only enables text entry with the above limitations, but also reserves the majority of screen real estates for the user interaction with physical reality. Regarding the touchscreen interaction, the nowadays smartphones offer users sufficient touch-sensitive areas to perform indirect manipulation on either well-defined buttons [55] or imaginary planes [52]. Forceboard [55] provides a thumb-size circular button for the text selection on a scanning keyboard containing 26 alphabets and the white space, in which the user's thumb press on a button will decide the character keys. By leveraging the well-known yet ingrained arrangement of QWERTY keyboard, users with Gesture Typing draw a trajectory passing through the keys inside an imaginary QWERTY keyboard without any visual cues [52].

To the best of our knowledge, the research community puts significant efforts on designing systems for character-level and word-level text entry on AR smartglasses. Another efficient way to achieve high-volume inputs of text, copy-and-paste operations, has been neglected. The existing works treat the textual contents in AR as vector graphics (e.g. BISHARE [56]), and their interaction limits to re-sizing and rotational movements of texts in the form of vector graphics[5]. BISHARE [56] investigates six primary design space of multi-display collaboration between AR headsets and smartphones, in which the smartphone can serve as an IMU-based ray-casting pointer interacting with iconic 3D text objects in mid-air. Therefore, copy-and-paste operations for texts in AR with the premise of managing various granularity are under-explored. The *direct manipulation* approaches, as discussed in Section 2.1 and 2.2, mainly focus on the *direct touches of textual contents shown on touchscreen interfaces, in which both the user's attention (gaze) and actions (fingers) are working on the identical screen interface.* In contrast, this paper addresses the copy-and-paste operations through *indirect manipulation* with AR smartglasses. By utilizing the smartphone touchscreens (as a test-bed [53]), characterized by responsive and accurate sensing capabilities, a miniaturized and circular interface is designed for interacting with textual contents for copy-and-paste operations. In addition, we further consider the nature of AR smartglasses designated for various mobile

[4] Mad Gaze Ares: https://www.madgaze.com/ares/

[5] AR Text Manipulation: https://dribbble.com/shots/4548785-AR-Text-Manipulation

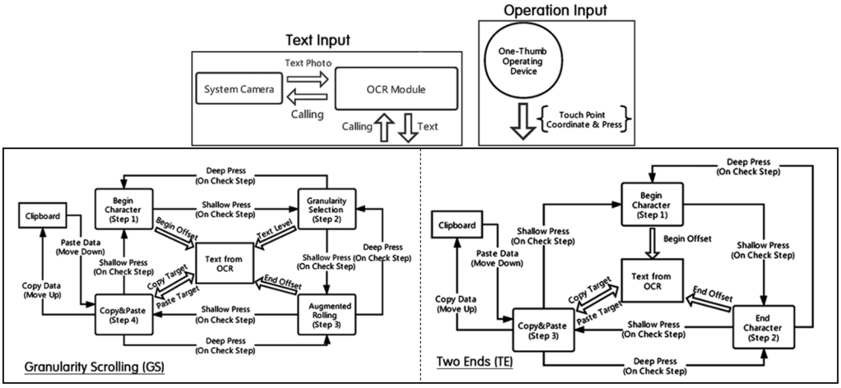

Operation Input
Text Input
Text Photo
System Camera
OCR Module
Calling
Calling
Text
One-Thumb Operating Device
Touch Point Coordinate & Press
Deep Press (On Check Step)
Begin Character (Step 1)
Shallow Press (On Check Step)
Granularity Selection (Step 2)
Clipboard
Begin Offset
Text Level
Paste Data (Move Down)
Text from OCR
Deep Press
Shallow Press (On Check Step)
Shallow Press (On Check Step)
Copy Target
End Offset
Copy Data (Move Up)
Paste Target
Copy&Paste (Step 4)
Shallow Press (On Check Step)
Augmented Rolling (Step 3)
Deep Press (On Check Step)
Granularity Scrolling (GS)
Deep Press (On Check Step)
Begin Character (Step 1)
Clipboard
Shallow Press (On Check Step)
Begin Offset
Shallow Press (On Check Step)
Paste Data (Move Down)
Text from OCR
Copy Target
End Offset
Copy Data (Move Up)
Paste Target
Copy&Paste (Step 3)
Shallow Press (On Check Step)
End Character (Step 2)
Two Ends (TE)
Deep Press (On Check Step)

Fig. 1. The two system solutions named **Granularity Scrolling (GS)** containing four major steps, and **Two Ends (TE)** consisting of three key steps.

scenarios. For the sake of mobility, our text selection and text relocation strategies are appropriately designed, which are bounded by a miniaturized interaction area supported by user evaluations in this paper. The studies of miniaturized interaction area are important, which prove the feasibility of highly mobile AR inputs. By concatenating our works with other miniaturized and circular interfaces for text entry [55] and drone flights [53] (an example of IoT devices), the interaction (input) coverage [25] with such miniaturized surfaces has been further expanded.

3 SYSTEM DESIGN

This section describes the system design of *Press-n-Paste (PnP)*. First, we give a system overview of the two proposed solutions (Section 3.1) with justifications (Section 3.2). Then, the interaction design, including the in-text caret navigation (Section 3.3) as well as the procedures of text selection (Section 3.4) and copy-and-paste (Section 3.5) are explained in details. Finally, the implemented system is highlighted in Section 3.6.

3.1 System Overviews

The two proposed system solutions (Figure 1) share a similar architecture, and we generalize the similarity of the solutions, as follows. On the one hand, the *system camera* embedded in the AR smartglasses enables users to take a picture containing textual contents. Accordingly, the captured picture is processed by the *Optical Character Recognition (OCR) Module*, in order to recognize the text objects in the picture and make the indirect manipulation possible among the recognized text objects. The embedded cameras are now the standard sensors on AR smartglasses [25]. On the other hand, we design a miniaturized interface that is a circular thumb-size button on a touch-sensitive and pressure-sensitive surface. Such surfaces are widely available on nowadays smartphones and smartwatches. The touch-sensitive property can determine the thumb movements and their relative positions, which supports the user's subtle gestures such as hold-and-rub and rotational pan inside the circular area of the button. Through the mapping of the level of pressure exerted by the user's thumb, the pressure-sensitive property can augment the richness of the user's touch in such a size-constrained button [39]. It is important to note that the thumb-size interface, with the intention of designing a highly mobile solution, cannot accommodate large movements of the user's thumb, and hence the pressure-sensitive property can play an important role of replacing such space-consuming gestures.

In this way, the button allows users to perform thumb-driven interaction with the recognized text objects in the physical environment. Among the two solutions, namely Granularity Scrolling (GS) and Two Ends (TE), their key difference appears in the approaches of indirect manipulations, primarily in the middle of the loops of the respective illustrations. As shown in Figure 1 (upper), GS is a four-step approach containing the identification of the beginning character, granularity selection, augmented rolling, and copy-and-paste. In Figure 1 (lower), TE takes two identical steps (step 1 and 2) to identify the beginning character and ending character of the target textual body. The interaction details of GS and TE will be explained in Section 3.3–3.5. After taking the respective steps for indirect manipulations of the target textual body, the selected texts are stored in the *clipboard*, and consequently are ready for paste operations to target locations in an in-text environment.

3.2 Justifications of the System Design

The copy-and-paste system has long been solved by direct manipulation solutions in research community and commercial products. Due to the diminishing touch area on AR smartglasses, this paper proposes two alternative solutions (GS and TE). We further consider the actual use cases of smartglasses and put user mobility as the first priority. Therefore, we strive to balance the usability and the user mobility by designing a miniaturized button on a surface leveraging two modals of touch and pressure [12]. We explain the chosen modals in the following.

On the one hand, touch input offers precise, tactile and comfortable user input [25], and outweighs speech commands by the high level of the sense of agency (i.e. the users feel more confident with the input modal) [29]. In addition, the text selection and relocation with voice-based inputs may cause usability issues. For instance, characters and words can be duplicated in a piece of written contents, and users with voice-based inputs cannot easily select specific characters or words. The text relocation can pose difficulties because indicating a particular target location in the text can be vague and ambiguous [9].

On the other hand, computer-vision methods supported by optical sensors have inherent drawbacks of detecting the emulated user's touches [11]. The mid-air scenarios under the egocentric view make the camera looking behind the tapping finger. The user's fingers are bigger than the characters that are small and densely packed, and thus the vision occlusion occurs. Thus, it is not easy to accurately detect when the mid-air finger contacts the digital overlays [28]. Second, computer-vision methods usually demand considerable computational resources and introduce a latency of variant length [26]. For instance, a most recent work, investigating mid-air touch sensing with depth cameras on the Microsoft Hololens [51], reports high rates of missed touches (3.5%) and spurious extra touches (19.0%), with a system latency of about 180 ms. Considering that every character is the candidate object to be selected, such latency can damage the user performance as the adjacent characters can be mistakenly chosen. In contrast, prior works have demonstrated the low interaction delay [19] and spatial accuracy [4] on the experience of touch-sensitive surfaces. Therefore, computer-vision contact sensing does not provide a satisfying solution for in-text interaction.

Both solutions employ multi-step approaches to accomplish indirect manipulations in the texts supported by the pressure as the second modal. Pressure is an emerging modal of user interactions in recent years, which has been employed in some smartphone solutions for user interactions, including controlling scroll bars [2], zooming in and out [48], text entry [55], augmented touch and multi-mode selection [12], drone flights [53], and reaching icons with firm grasps [7, 8]. These examples demonstrate high levels of usability in the pressure as an input modality. Also, its key benefits, such as subtle movements with thumb-based interaction, one-handed interaction, and space-saving interfaces [27] fit into our objective of mobile usage. Our proposed thumb-size button

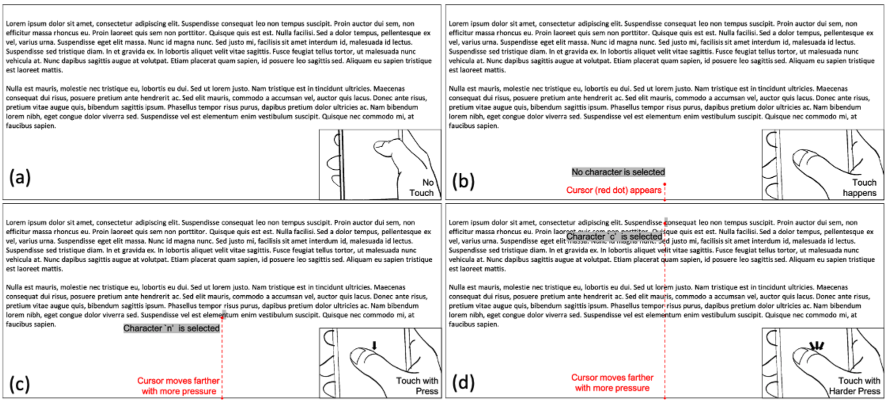

(a)
No Touch
(b)
No character is selected
Cursor (red dot) appears
Touch happens
(c)
Character 'n' is selected
Cursor moves farther with more pressure
Touch with Press
(d)
Character 'c' is selected
Cursor moves farther with more pressure
Touch with Harder Press

Fig. 2. Varying the caret reaches by exerted pressures: (a) without Touch; (b) thumb touches on the screen and then perform press in (c) and harder press in (d).

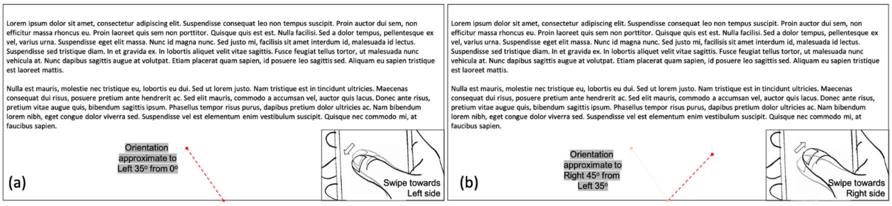
also leverages the pressure to shrunken the interface size as minimal as possible [27, 53, 55]. It is worth noting that, by considering the prior works leveraging pressure-sensitive interaction within button-size interface, such minimal interfaces can be employed on a spectacle frame of smartglasses or a smart ring, which avoid bulky handheld devices and allow two unoccupied hands in daily routine tasks. The rest of the paragraphs discusses the user interactions in two interaction solutions in details. It is important to note that, throughout these paragraphs, the pressure modality plays an important role in distinguishing the interaction procedures, caret navigation, and granularity management.

3.3 Caret Navigation

On touchscreen interfaces, it is difficult to place the caret precisely on the text in small devices. The user's finger first taps on a text, but the caret usually appears on incorrect places and is near to the target so that they need to slide their finger on the screen, but the targeting text is usually under the finger, due to the fat finger problem [47]. Apart from the problematic interaction, smartglasses users frequently switch their gazes (up and down) and attention between multiple displays, and the usability, therefore, deteriorates the user performance [41], in case that the textual contents directly shown on the touchscreen. In order to achieve aligned user attentions, the proposed system makes the caret navigation on the heads-up display of smartglasses.

The indirect manipulation of in-text caret navigation is driven by two modals: touch coordinates and pressure. Figures 2 and 3 describe the reachability and orientation of the in-text caret respectively, in which the texts are shown in a heads-up display of AR smartglasses (16:9 ratio), and the thumb-based interactions on a touchscreen surface (pressure-sensitive as well) is correspondingly demonstrated. In a high-level description, users can employ one-handed thumb presses with varying pressure level and touch coordinates to reach any characters in the text environment.

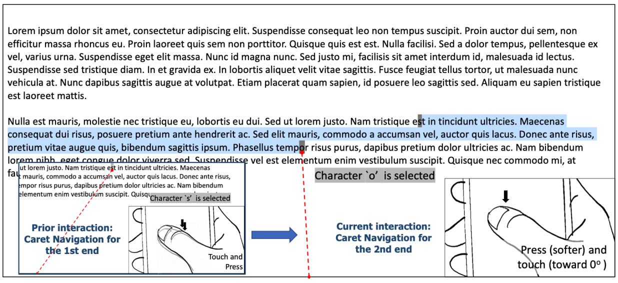
In Figure 2, the caret initially does not appear (a) until any touch event is detected (i.e. users tap on the touchscreen). Afterwards, the user's thumb locates on the touchscreen and the caret pops-up from the bottom of the heads-up display (b). The extent of the caret reach is directly proportional to the pressure level exerted by the user's thumb. Therefore, a tap on the screen, generating the weights of the thumb, makes a very short reach. As long as the user exerts more pressure on the touchscreen, the caret reaches further to a character 'n' at the last line of the second paragraph (c). As the user makes the harder press, the caret can extend to the upper edge area and hence select a character 'c' (d). In Figure 3, similarly, the dotted lines in colours of pale pink and red indicate


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse consequat leo non tempus suscipit. Proin auctor dui sem, non efficitur massa rhoncus eu. Proin laoreet quis sem non porttitor. Quisque quis est est. Nulla facilisi. Sed a dolor tempus, pellentesque ex vel, varius urna. Suspendisse eget elit massa. Nunc id magna nunc. Sed justo mi, facilisis sit amet interdum id, malesuada id lectus. Suspendisse sed tristique diam. In et gravida ex. In lobortis aliquet velit vitae sagittis. Fusce feugiat tellus tortor, ut malesuada nunc vehicula at. Nunc dapibus sagittis augue at volutpat. Etiam placerat quam sapien, id posuere leo sagittis sed. Aliquam eu sapien tristique est laoreet mattis.

Nulla est mauris, molestie nec tristique eu, lobortis eu dui. Sed ut lorem justo. Nam tristique est in tincidunt ultricies. Maecenas consequat dui risus, posuere pretium ante hendrerit ac. Sed elit mauris, commodo a accumsan vel, auctor quis lacus. Donec ante risus, pretium vitae augue quis, bibendum sagittis ipsum. Phasellus tempor risus purus, dapibus pretium dolor ultricies ac. Nam bibendum lorem nibh, eget congue dolor viverra sed. Suspendisse vel est elementum enim vestibulum suscipit. Quisque nec commodo mi, at faucibus sapien.

Orientation approximate to Left 35º from 0º
Swipe towards Left side
(a)
Orientation approximate to Right 45º from Left 35º
Swipe towards Right side
(b)

Fig. 3. Controlling the caret orientations by altering the touch coordinates: the thumb swipes/pans toward (a) left hand side and (b) right hand side.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse consequat leo non tempus suscipit. Proin auctor dui sem, non efficitur massa rhoncus eu. Proin laoreet quis sem non porttitor. Quisque quis est est. Nulla facilisi. Sed a dolor tempus, pellentesque ex vel, varius urna. Suspendisse eget elit massa. Nunc id magna nunc. Sed justo mi, facilisis sit amet interdum id, malesuada id lectus. Suspendisse sed tristique diam. In et gravida ex. In lobortis aliquet velit vitae sagittis. Fusce feugiat tellus tortor, ut malesuada nunc vehicula at. Nunc dapibus sagittis augue at volutpat. Etiam placerat quam sapien, id posuere leo sagittis sed. Aliquam eu sapien tristique est laoreet mattis.

Nulla est mauris, molestie nec tristique eu, lobortis eu dui. Sed ut lorem justo. Nam tristique est in tincidunt ultricies. Maecenas consequat dui risus, posuere pretium ante hendrerit ac. Sed elit mauris, commodo a accumsan vel, auctor quis lacus. Donec ante risus, pretium vitae augue quis, bibendum sagittis ipsum. Phasellus tempor risus purus, dapibus pretium dolor ultricies ac. Nam bibendum lorem nibh, eget congue dolor viverra sed. Suspendisse vel est elementum enim vestibulum suscipit. Quisque nec commodo mi, at fau

ut lorem justo. Nam tristique est in tincidunt ultricies. Maecenas
t mauris, commodo a accumsan vel, auctor quis lacus. Donec ante risus,
empor risus purus, dapibus pretium dolor ultricies ac. Nam bibendum
elementum enim vestibulum suscipit. Quisqu

Character `s' is selected
Prior interaction: Caret Navigation for the 1st end
Touch and Press
Character `o' is selected
Current interaction: Caret Navigation for the 2nd end
Press (softer) and touch (toward 0º)

Fig. 4. A pictorial description of the TE solution by identifying the first end (the inside figure) and the second end.

the original and resultant caret orientations, respectively. The caret orientation matches with the direction changes of touch coordinates. The user's thumb shifts slightly to the left-hand side and the caret orientation changes to left accordingly (a). In contrast, a moving thumb towards right drives the caret to the right-hand side (b).

3.4 Multi-step Text Selection

This section explains the multi-step approaches of GS and TE. First, we clarify the confirmation methods of the touch and pressure-based gestures. Through detecting 2D touch coordinates, gestures including *Orientation Controls in Caret navigation*, Up and Down Swipes, (Anti-)Clockwise Scrolling, can be recognized in an instant and unambiguous manner. However, pressure modality is transient in a continuous range. And a confirmation technique is necessary to finalize the user inputs. The most common confirmation techniques are Dwell Time and Quick Release [39]. Due to its reliability (97%) [8], we apply a dwell time of 500 ms, for *Press in Caret navigation*, *Shallow Press*, *Deep Press*, and *Granularity Management*. For example, when the caret holds at a certain character for 500ms, the character will be selected. In the multi-step loops as shown in Figure 1, the decision made in one step will proceed to another step if the user confirms the selection by doing *Shallow Press*, while the incorrect decision can be erased by performing *Deep Press*. Both the *Shallow Press* and *Deep Press* are user's presses, but they are differentiated by a normalized pressure threshold of 0.5. Users can employ either light presses or normal taps to do the *Shallow Press*. And *Deep Press* are regarded as intended strong presses.

The solution named Two Ends (TE) contains two identical steps of caret navigation to select two characters representing two ends of the target texts. Figure 4 depicts the selected texts between the first selected character 's' and the second selected character 'o' (inclusive). The granularity management in TE is flexible and largely depends on the selection of two characters, regardless of the selection order. Here is a supplementary example – if the first paragraph is the target texts, the character 'L' at the beginning and the character (symbol) '.' at the end should be selected.

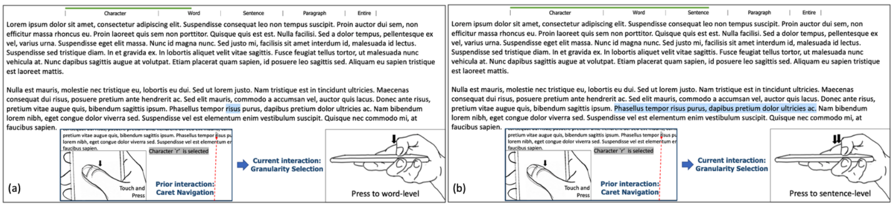

Character
Word
Sentence
Paragraph
Entire
Character 'r' is selected
Touch and Press
Prior interaction: Caret Navigation
Current interaction: Granularity Selection
Press to word-level
Press to sentence-level
(a)
(b)

Fig. 5. A pictorial description of the granularity management driven by the level of pressure being exerted: (a) word-level (b) sentence-level.

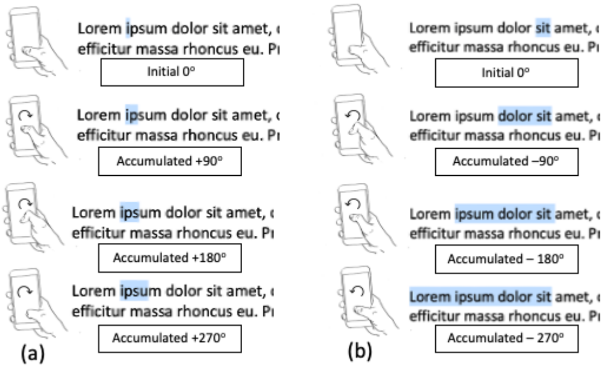
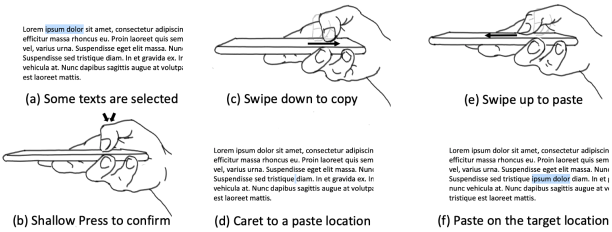


Fig. 6. A pictorial description of the scrolling gestures after the granularity is defined: (a) clockwise rotation for character-level selection (b) anti-clockwise rotation for word-level selection.

Instead of taking two caret navigation, Granularity Scrolling (GS) firstly take only one caret navigation to reach the beginning character, and secondly, a widget designated for granularity management exists on the top edge of the screen for the purpose of screen spacing-saving [28]. Considering the chosen character in Section 3.3 as the origin, the user's thumb employs the right level of pressure to select the granularity, including *Character*, *Word*, *Sentence*, *Paragraph* as well as the *Entire* document. Two examples are shown in Figure 5, as follows. After a character 'r' is chosen, the user decides the pressure (**a green bar as visual cues**) reaching the granularity of 'Word' and holds a dwell time of 500ms. Accordingly, the word 'risus' is selected (a). If the granularity of 'Sentence' is chosen, the whole sentence is selected (b). After confirming the granularity, the third step is to perform scrolling gestures to decide the unit numbers of texts at the chosen granularity. A 90° scrolling gesture (i.e. 360° = four more units) in either clockwise or anti-clockwise direction will pick the next forward unit or backward unit on the basis of the selected text at the chosen granularity, respectively. Over-scrolling can be reverted by doing scrolling in the opposite direction. Figure 6a is an illustration of clockwise scrolling at character-level granularity. Throughout the scrolling from 0° to 270°, the number of characters, increasing one by one, changes from one initially to four characters eventually. Also, Figure 6b demonstrates an anti-clockwise scrolling at the word-level granularity. After 270° anti-clockwise scrolling, the adjacent three backward words are selected.

3.5 Copy and Paste

Figure 7 describes the final stage of *Press-n-Paste*. After the target texts are selected (a) by either GS or TE, the user confirms the selected texts by performing a shallow press (b). After the shallow press, the final step of copy and paste begins. We try to facilitate the user affordance [14] by introducing



Lorem ipsum dolor sit amet, consectetur adipiscin
efficitur massa rhoncus eu. Proin laoreet quis sem
vel, varius urna. Suspendisse eget elit massa. Nun
Suspendisse sed tristique diam. In et gravida ex. Ir
vehicula at. Nunc dapibus sagittis augue at volutp
est laoreet mattis.

(a) Some texts are selected (c) Swipe down to copy (e) Swipe up to paste

Lorem ipsum dolor sit amet, consectetur adipiscin
efficitur massa rhoncus eu. Proin laoreet quis sem
vel, varius urna. Suspendisse eget elit massa. Nun
Suspendisse sed tristique diam. In et gravida ex. In
vehicula at. Nunc dapibus sagittis augue at volutp
est laoreet mattis.

Lorem ipsum dolor sit amet, consectetur adipiscin
efficitur massa rhoncus eu. Proin laoreet quis sem
vel, varius urna. Suspendisse eget elit massa. Nun
Suspendisse sed tristique ipsum dolor diam. In et
nunc vehicula at. Nunc dapibus sagittis augue at v
tristique est laoreet mattis.

(b) Shallow Press to confirm (d) Caret to a paste location (f) Paste on the target location

Fig. 7. A pictorial description of the copy-and-paste operations.

intuitive gestures in the copy and paste operations, where swiping down and up emulates the dragging and shooting of the selected texts. Therefore, the user's thumb swipes down to copy the selected texts (c). Afterwards, the user selects a target paste location with the caret navigation (d). Finally, the user performs a swipe gestures upward (e) to accomplish the paste operation (f).

3.6 System Implementation

We implemented an experimental prototype named Press-n-Paste (PnP) on a Microsoft HoloLens (version 1) and an iPhone 7 (Figure 8), which are implemented as a web-based application in JavaScript, HTML and CSS. To implement the gestures as described in Section 3.4 and 3.5, GS and TE leverage the touchscreen on an iPhone 7 that are both touch-sensitive and pressure-sensitive surfaces. The iPhone 7 as a controller enables users to interact with the text contents projected on the heads-up display inside the Microsoft Hololens. Also, both the iPhone 7 and Microsoft Hololens communicate via a web page in the browser through WebSocket. The communication flow is primarily from the iPhone 7 to the Microsoft Hololens, in which the iPhone 7 sends the sensor data to the Microsoft Hololens such as (x, y) touch coordinates and Normalize Pressure Level (NPL) ranging from 0 to 1. The normalized range is converted from a scale between 0 – 6.667 unit in the default setting of iPhone 3D touch. As reported by an iOS application named *Digital Scale+*[6], the press-sensitive surface on iPhone 7 can detect object weights up to 385 grams, which is equivalent to an approximate value of 3.78 N (Newton). Meanwhile, the Microsoft Hololens responds to the user gestures (e.g. shallow press, deep press, scrolling, and swipes), and materializes the visuals of pressure widgets such as the caret navigation and granularity management. For example, the projection of caret position is computed by the sensor data, and accordingly, the caret rendering follows.

According to Figure 1, the system begins with capturing an image thought the system camera on the Microsoft Hololens. The standard image-to-text module in Optical Character Recognition (OCR) converts the texts from an image. Accordingly, the entire texts in the image form will become text objects to be manipulated on the Microsoft Hololens. Throughout the multi-step solutions of GS and TE (Figure 9) and TE (Figure 10), the step transitions are governed by either shallow press (confirmation) or deep press (revert). As mentioned in Section 3.4, if a shallow press is exerted (NPL <= 0.5), the selected item will proceed to the next step. Otherwise, the selected item will reset when a deep press is detected (NPL > 0.5). Users can easily distinguish between two divisions of pressure-sensitive taps [27]. In addition, scrolling and swipe gestures are solely supported by the touch coordinates on the iPhone 7, while the caret navigation (subtle pan gestures and pressure) and granularity management (tap and pressure) utilize the two sensor data of pressure levels and touch coordinates on an iPhone 7. The pressure sensors on iPhone 7 can smoothly manage the continuous spectrum of exerted pressure levels, where prior work [53] demonstrates drone movements without jerking with a linear function. Also, filtering techniques are necessary for DIY-sensing units [27]

[6] Digital Scale+: https://github.com/wernjie/digital-scale-plus

but not necessary on iPhone 7. Thus, we solely employ *pressure.js*[7] without any filtering techniques to detect pressure-sensitive gestures between the web application and the sensors in every 2.1 ms. On the other hand, the exerted pressure level is linearly mapped to the NPL, as the linear mapping achieves reasonable levels of usability [33, 38, 50]. Non-linear functions pose usability issues: the parabolic-sigmoid function results in slower response at the two ends of the pressure spectrum [40]. And the quadratic function introduces the user's cognitive loads and learning costs to the uneven pressure distribution (i.e. larger pressure at the lower end, and smaller pressure at the higher end). More implementation details of caret navigation and granularity management are as follows.

- **Caret navigation**: A minimum threshold value of 0.2 is defined to avoid unintended presses [53]. With the linear mapping, the NPL value between $0.2 - 1.0$ maps to the caret position along the projected line, and the radius (max. pressure, 1.0) is originated from the mid-point of the bottom edge to one of the upper corners. If the NPL value goes beyond the screen area, the caret will be bounded within the screen edges, for instance, e.g. the caret will stay at the mid-point of the upper edge even the NPL exceeds 0.9. Additionally, the subtle pan across several pixels (< 3 mm) indicates the orientation of the projected line, and the subtle movement does not significantly impact the pressure control [53], especially when the circular button, located at the bottom bezel (ϕ10.5 mm centred at (x = 187.5,y = 475.5)) on the smartphone, is fully enclosed by the actionable thumb ranges of the metacarpophalangeal (MCP) joint. In other words, the subtle panning with a button at the bottom bezel will not trigger the muscle constraints at the MCP joint and hence considerable pressure variation, so the users can do stable caret navigation. Furthermore, as discussed in [53], the form-factor difference between smartphone and ring-form devices only influences the performance time due to the device weights, but not the subtleness of the interaction techniques. In this way, the caret can sweep through any location along the projected line, and get the target character through Web API *ElementsFormPoint* [8].
- **Granularity management**: The NPL is divided into a ratio of 4:2:2:2:1 for the five textual granularities of character, word, sentence, paragraph, and the entire document. We allocate sufficient space to the character level to reduce the chance of overshooting to higher levels, as the thumb weights without intentionally presses can produce pressure on the screen. Prior work shows that users with sufficient feedback cues can distinguish up to 10 options in a pressure-sensitive menu [50]. Thus, five options in a menu demonstrate a reasonable usability. Additionally, a pop-up pie menu is employed in a similar approach for direct manipulation on touchscreen [10]. However, we employ a linear menu at the upper edge to reserve the central area of the limited screen real estate on AR smartglasses [28].

4 EVALUATION I: USABILITY STUDY

We assess the user performance of GS and TE for the copy-and-paste operations in the experimental prototype (PnP). We recruited 18 participants (18 - 33 years ($\overline{M}$ = 25.94, σ=4.22); 16 male and 2 female; 17 right-handed) from a university campus. All of them are experienced smartphone users but have no prior experience with pressure widgets. Each participant attended an approximately 65-minute experiment. All participants completed eight sessions for each solution, and a total of 725,560 touchpoints for GS and TE were tracked. A 15-minute warm-up session [10] was allowed before the sessions. At the beginning of each session, all participants were told in each session to complete the task as fast as possible. They can only correct their mistakes for the current sub-tasks. After all the sessions, a NASA Task Load Index (NASA TLX) questionnaire [16] was distributed

[7] Pressure.js: https://pressurejs.com/index.html

[8] ElementsFormPoint: https://developer.mozilla.org/zh-CN/docs/Web/API/Document/elementsFromPoint

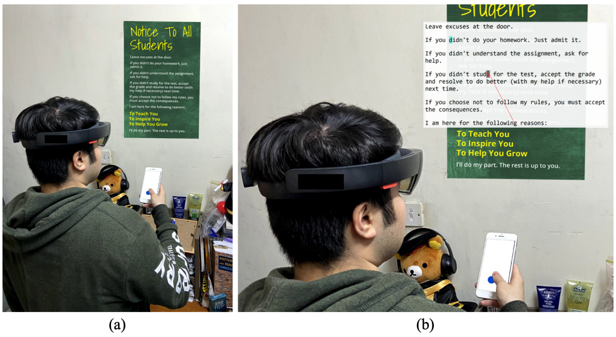

Notice To All Students
Students
Leave excuses at the door.
If you didn't do your homework. Just admit it.
If you didn't understand the assignment, ask for help.
If you didn't study for the test, accept the grade and resolve to do better (with my help if necessary) next time.
If you choose not to follow my rules, you must accept the consequences.
I am here for the following reasons:
To Teach You
To Inspire You
To Help You Grow
I'll do my part. The rest is up to you.
(a)
(b)

Fig. 8. The setting of experimental prototype with a Microsoft Hololens and an iPhone 7, in front of a poster in (a) to illustrate the capture of real-life textual contents in the user's surroundings; The small-sized circular button on the iPhone 7 serves as an indicative press location. When the user focuses on the heads-up display in (b), the user's thumb presses can act outside the button area, for which the frequent switch of attention between the heads-up display and smartphone touchscreen can be minimized.

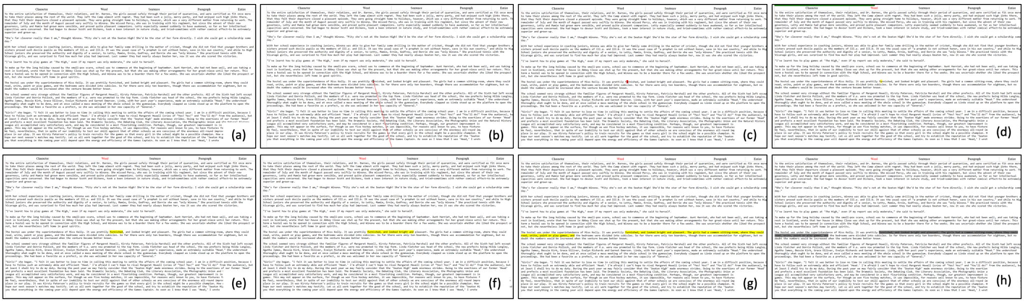


Fig. 9. An example of the GS interface in the experimental prototype during the text selection stage: (a) default screen with no user interaction; (b) caret navigation on the target character; (c) a character is selected after the dwell time and further confirmed with a shallow press; (d) selecting word-level granularity in the granularity management; (e) selecting the beginning word; (f) after the shallow press, scrolling gestures in the clockwise direction; (g) more scrolling gestures in the clockwise direction; (h) after the shallow press, the selected text is ready to the final stage of the copy-and-paste operation.

to every participant to understand the qualitative metrics of the user workloads about the two solutions (GS and TE), in terms of Mental, Physical, Temporal, Performance, Effort as well as Frustration. The usability study was carried out complying the General Data Protection Regulation (GDPR) and approved by university institutional ethics review board (IRB) regulations.

The usability study aims to investigate the user behaviour and user performance of the two proposed solutions under various granularities. Participants in a seating posture hold an iPhone 7 to accomplish the evaluation with the two proposed solutions (GS and TE) in the experimental setting (Figure 8), as discussed in Section 3.6. The visual cues of the two solutions are shown on the AR display. The 18-pt font size [6] is applied in all sessions. Due to some participants cannot read the contents clearly on the mobile headset [31], a distal 16:9 screen display on a 15" monitor (1 meter away from the user's seat) was alternatively used to emulate the Microsoft Hololens display, ensuring the participants to read the dense and small textual contents with no difficulties. It is important to note that the caret navigation and granularity management employ a scalar projection from the mid-point of the bottom edge, and the pixels inside the screen doesn't impact the user performance with the pressure-sensitive interactions.

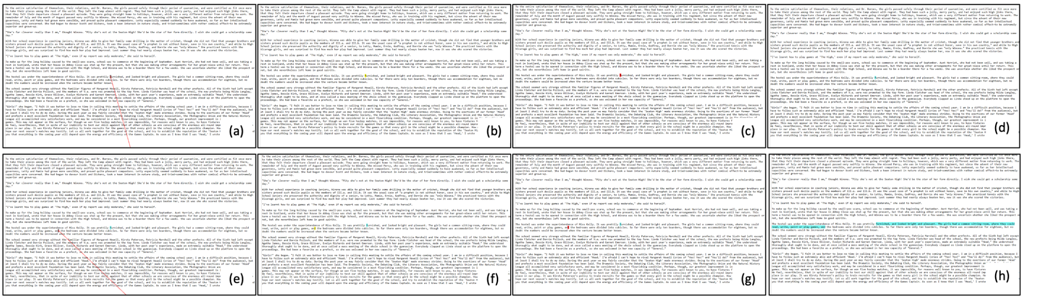


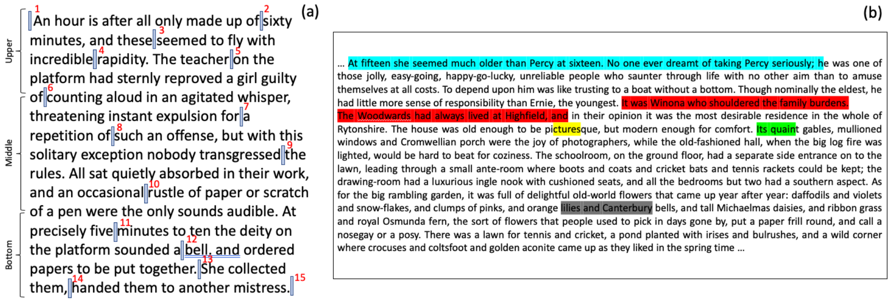
Fig. 10. An example of the TE interface in the experimental prototype during the text selection stage. (a) the first caret navigation on the target character; (b) a character is selected after the dwell time; (c) a shallow press to confirm the selected character; (d) a state with visual clue ready for the next caret navigation; (e) the second caret navigation on the target character; (h) as (b); (g) as (c); (h) the target text between the two caret positions is selected.

On the smartphone, a circular button of ϕ10.5 mm centred at (x,y) coordinate = (187.5, 475.5) serves to guide the suggested interaction surface. Every session consists of nine tasks considering full coverage of granularities [10], as follows : (1) Sub-word as 6 characters inside a word; (2) Word and characters as 1 words and 5 characters; (3) 3 words; (4) 1 sentence; (5) 1 paragraph; (6) 2 sentences and 1 characters; (7) 1 sentences and (across paragraph) 8 words; (8) 2 sentences (separated by two paragraphs); (9) Entire text. All these sub-tasks exist in a two-paragraphs text (Figure 11b). After selecting the text in a sub-task, the participants were asked to copy and paste the selected text in a text box containing evenly distributed paste locations (Figure 11a). The design of nine tasks is modified from the prior work [10], in which tasks (1 − 5) and (8 − 9) hold the same granularity to understand the performance of techniques at various textual granularity. In [10], their tasks of selecting (multi-)paragraph(s) are resembling to the operations at other granularities, and hence we replace them by tasks 6 and 7 in our study. Our new tasks (6 & 7) present more comprehensive views on the cross-granularity text selection (i.e. dealing with small pieces of texts in physical surroundings), where their only task of cross-granularity text selection, 'char to paragraph' in [10], shows no advantages than the commercial solutions. The text material is extracted from an electronic library platform named Gutenberg[9], which is an easy-to-read text for children and a reasonable mock-up of daily reading material. Finally, all interaction approaches (GS and TE) and tasks (1 − 9) are counter-balanced to minimize the carry-over effects that potentially hurt the internal validity of our findings.

4.1 User Performance

We evaluate the quantitative performance of nine-task completion times and error rates in all sessions in the within-subject evaluation. We record a total of 1,296 copy-and-paste operations (18 participants * 9 tasks * 8 sessions). On the one hand, GS and TE achieve the overall completion times (sum average of the nine task) of 193,074 ms (σ=60,175 ms) and 163,148 ms (σ=51,010 ms) for the 9 tasks at various granularity. Additionally, we separate the overall time of GS and TE into two major time components that are text selection (Section 3.3– 3.4) and copy-and-paste operations (Section 3.5). Regarding the text selection components, two-way RM-ANOVA demonstrates a significant effect of the Interaction Approaches and the Sessions ($F_{1,7}$ = 54.48, p <0.01), indicating the significance of interaction approaches on the completion time and the learning effect between

[9] The Project Gutenberg eBook, The Luckiest Girl in the School, written by Angela Brazil," https://www.gutenberg.org/files/18019/18019-h/18019-h.htm



An hour is after all only made up of sixty minutes, and these seemed to fly with incredible rapidity. The teacher on the platform had sternly reproved a girl guilty of counting aloud in an agitated whisper, threatening instant expulsion for a repetition of such an offense, but with this solitary exception nobody transgressed the rules. All sat quietly absorbed in their work, and an occasional rustle of paper or scratch of a pen were the only sounds audible. At precisely five minutes to ten the deity on the platform sounded a bell, and ordered papers to be put together. She collected them, handed them to another mistress.

… At fifteen she seemed much older than Percy at sixteen. No one ever dreamt of taking Percy seriously; he was one of those jolly, easy-going, happy-go-lucky, unreliable people who saunter through life with no other aim than to amuse themselves at all costs. To depend upon him was like trusting to a boat without a bottom. Though nominally the eldest, he had little more sense of responsibility than Ernie, the youngest. It was Winona who shouldered the family burdens.
The Woodwards had always lived at Highfield, and in their opinion it was the most desirable residence in the whole of Rytonshire. The house was old enough to be picturesque, but modern enough for comfort. Its quaint gables, mullioned windows and Cromwellian porch were the joy of photographers, while the old-fashioned hall, when the big log fire was lighted, would be hard to beat for coziness. The schoolroom, on the ground floor, had a separate side entrance on to the lawn, leading through a small ante-room where boots and coats and cricket bats and tennis rackets could be kept; the drawing-room had a luxurious ingle nook with cushioned seats, and all the bedrooms but two had a southern aspect. As for the big rambling garden, it was full of delightful old-world flowers that came up year after year: daffodils and violets and snow-flakes, and clumps of pinks, and orange lilies and Canterbury bells, and tall Michaelmas daisies, and ribbon grass and royal Osmunda fern, the sort of flowers that people used to pick in days gone by, put a paper frill round, and call a nosegay or a posy. There was a lawn for tennis and cricket, a pond planted with irises and bulrushes, and a wild corner where crocuses and coltsfoot and golden aconite came up as they liked in the spring time …

Fig. 11. (a) An example task interface of the 15 paste locations, in which the target location will be highlighted; (b) A sample task interface for the text selection – Pale Blue (task 6): 2 sentences and 1 characters, Red (task 7): 1 sentences and across paragraph 8 words, Yellow (task 1): Sub-word as 6 characters inside a word, Green (task 2): Word and characters as 1 words and 5 characters, and Grey (task 3) 3 words.

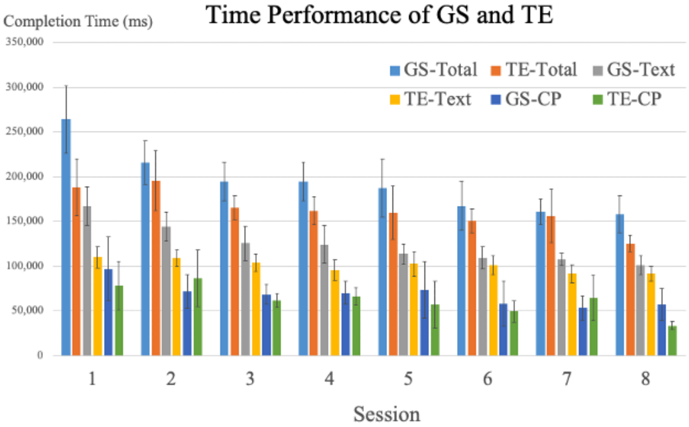


Fig. 12. Performance Times (sum average of nine tasks) of GS and TE, where the error bars are the values of standard deviations. After 8 sessions, the users is able to achieve a practical CP operations (TE, 13,951 ms), where two reference commercial solutions with direct manipulation on touchscreens such as long-tap and 2-Fingers are 14,260 ms and 16,810 ms, respectively [6].

sessions among the text selection conditions of GS and TE. At the stage of text selections, the participants with TE (mean=100,871 ms, σ= 22,153 ms) complete the tasks 23.28% faster than GS (mean=124,358 ms, σ=36,966 ms). During the sessions 1 – 4 of text selections, we observe a significant performance gap between both methods. The 3-step GS poses higher complexity than TE, causing the participants to take a longer time to get familiar with GS. As shown in Figure 12, the average completion times for GS(-Text, grey bars) and TE(-Text, yellow bars) decrease by 65.56% and 20.07% throughout the eight sessions. At the final session, the participants with GS (mean=101,073 ms, σ=20,925 ms) and TE (mean=91,868 ms, σ=16,849 ms) improve their text selection time from the first session, in which GS and TE result in 167,334 ms (σ= 43,215 ms) and 110,305 ms (σ= 23,860 ms) respectively.

However, two-way RM-ANOVA shows no effect of the interaction approaches on the overall completion time and the learning effect ($F_{1,7}$ = 1.61, p= 0.21) between sessions among the copy-and-paste selection conditions of GS(-CP, blue bar) and TE(-CP, green bar). As expected, both the GS (mean=68,722 ms, σ=45,801 ms) and TE (mean=62,277 ms, σ=42,824 ms) employ the same approaches using up and down swipes as well as caret navigations. In the final session, the CP operation times reduce to 57,096 ms (σ= 9,085 ms) and 33,693 ms (σ= 36,252 ms) for GS and TE, respectively. TE is sightly better than GS because the participants with TE were doing the caret navigation primarily throughout all the sessions.

Regarding the time component of text selection, we further analyze the effects of text granularity among the nine tasks to the selection time of GS and TE. Figure 13 depicts the selection time of GS and TE (sum average of eight sessions) in the nine tasks representing various text granularity. We observe that GS results in higher variation in completion times than TE among the nine tasks. We ran the one-way ANOVA to assess the effects of text granularity in each text selection conditions. Statistical significance exists in GS ($F_{8,153}$ = 19.89, p <0.01) but not TE ($F_{8,153}$ = 1.15, p = 0.33). Furthermore, we examine the statistical significance among tasks in GS and TE with a post-hoc analysis named Tukey's Honest Significant Difference test (Table 1). We omit the explanation of the post-hoc analysis for TE because of the absence of statistical significance, and reserve the discussion for GS as follows. On the whole, we generalize the statistical significance by the intersection between chosen granularity and the degree of scrolling gestures. First, the chosen granularity in GS impact the task completion time, as follows: Character-level (Tasks 1, 2 and 6), word-level (task 3 and 7) sentence-level (tasks 4 and 8), paragraph-level (task 5) granularities as well as the entire text (task 9). In addition, the circular degrees of scrolling gestures lead to different task completion time – Tasks 1, 2, 6, 7 need rigorous scrolling gestures more than 360°; tasks 3 and 8 require minimal scrolling gestures within 360°; tasks 4, 5, and 9 are accomplished without scrolling gestures, and the default unit captured in the caret navigation step provides the correct unit (one). A violin plot (Figure 14) is plotted to reinforce our statements, and correspondingly another one-way ANOVA examines the effect of steps to the completion time demonstrating the statistical significance ($F_{4,805}$ = 101.65, p <0.01). Among the three steps in TE, caret navigation, granularity management and scrolling gesture results in average unit operation times of 6,769 ms (σ = 2,685 ms), 3,250 ms (σ = 1,211 ms), and 3,798 ms (σ = 2,825 ms), respectively. From the third step of scrolling gestures demonstrating a high standard deviation value, GS is subject to the circular degree capturing the right amount of text units at the chosen granularity, although the third step only consumes 27.49% of the total time for text selection. The above findings also explain why GS (13,817 ms per text selection), modified from a state-of-the-art solution named *ForceSelect* for direct manipulation on touchscreens (7,100 ms per text selection) [10], is less competitive than TE (11,208 ms per text selection), under the setting of indirect manipulation for AR smartglasses.

Figure 14 depicts another important finding regarding the caret navigation. From the two identical steps in TE, the first and second steps of caret navigations in TE obtain average unit operation times of 6,797 ms (σ = 1,848 ms) and 4,411 ms (σ = 1,506 ms). Coincidentally, the first caret navigation steps in TE and GS achieve a very close mean value but distinct σ value. Also, we spot the time performance gap between the second caret navigation step in TE and the first caret navigation step in both GS and TE (2,358 – 2,386 ms). Our observations throughout the sessions find that the time overheads in the first steps can be explained by the cognitive process of planning the text capturing strategy, and the task nature causes the difference in σ values. In TE, the participants only require identifying the two ends of the textual body. However, the participants with GS encounters more diversified plans driven by both the caret positioning and the text granularity.

On the other hand, we recorded the number of errors in each session. Again, we separate the error counts into two halves as same as the one given in the completion time. Regarding the text selections, two-way RM-ANOVA shows a significant effect of the Interaction Approaches and the Sessions ($F_{1,7}$ = 23.81, p <0.01), which indicates the significance of interaction approaches on the error rate and the learning effect between sessions. In Figure 15, the blue and orange lines depict the average error rate for GS and TE over the eight sessions. The participants with GS

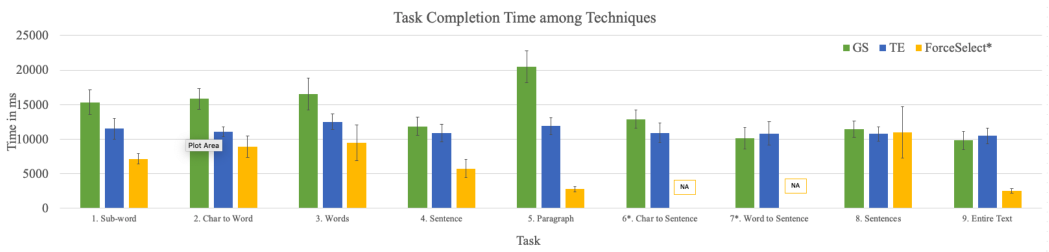


Fig. 13. Task completion times of text selection techniques of GS (Green) and TE (Blue) in the nine tasks under the setting of indirect manipulation, with the additional reference of a direct manipulation technique (*ForceSelect, FS*) (Yellow), where tasks 6 and 7 are not available (NA) in [10]. In general, the big performance gap exists between two similar approaches of GS and FS due to the difference in the direct and indirect manipulation. In contrast, TE in the simpler tasks of single-unit selection (4, 5 and 9) still maintains unavoidable overheads from the two caret navigation, and the gaps narrow down in more complicated tasks of multi-unit selection (1, 2, 3 and 8), as FS involves meticulous finger movements on a touchscreen.

Table 1. One-way ANOVA with Tukey HSD for GS (right upper corner) and TE (left lower corner) between two pairs of tasks, where bold numbers indicate statistical significance exists in the pairs

Task\Task	1	2	3	4	5	6	7	8	9
1	-	0.9000	0.9000	0.0508	**0.0010**	0.3984	**0.0169**	**0.0169**	**0.0010**
2	0.9000	-	0.9000	**0.0131**	**0.0016**	0.1688	**0.0037**	**0.0037**	**0.0010**
3	0.9000	0.7281	-	**0.0014**	**0.0149**	**0.0320**	**0.0010**	**0.0010**	**0.0010**
4	0.9000	0.9000	0.6009	-	**0.0010**	0.9000	0.8044	0.9000	0.6388
5	0.9000	0.9000	0.9000	0.9000	-	**0.0010**	**0.0010**	**0.0010**	**0.0010**
6	0.9000	0.9000	0.6132	0.9000	0.9000	-	0.2532	0.9000	0.1389
7	0.9000	0.9000	0.5517	0.9000	0.9000	0.9000	-	0.9000	0.9000
8	0.9000	0.9000	0.5026	0.9000	0.9000	0.9000	0.9000	-	0.8563
9	0.9000	0.9000	0.2930	0.9000	0.7466	0.9000	0.9000	0.9000	-

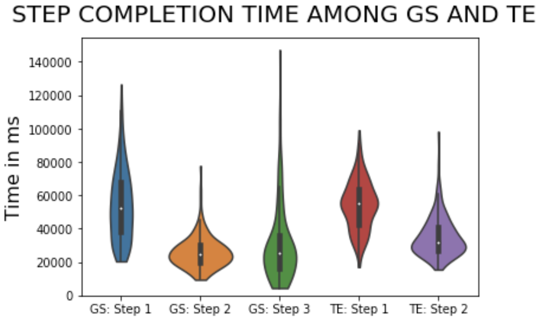


Fig. 14. A violin graph showing the population density of the completion times for the separate steps in GS and TE.

(mean=12.11%, σ=0.12) complete the tasks 6.32% more erroneous than TE (mean=5.79%, σ=0.11). At the first session, GS and TE result in the error rates of 19.14% (σ=0.12) and 3.70%. During the seventh session, the participants with TE reach the lowest error rates of 1.85% (σ=0.06), while GS reaches the lowest error rates of 6.79% (σ=0.09) at the sixth session. We notice that the user

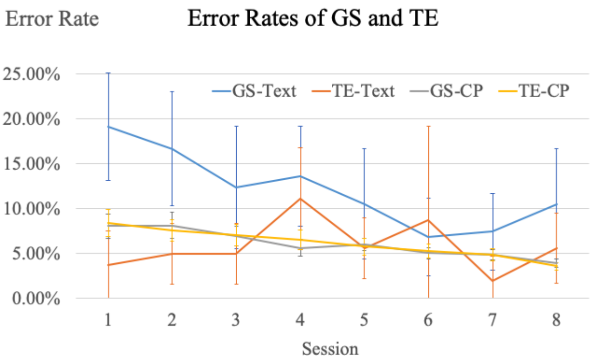


Fig. 15. Error rates of GS and TE across sessions, where the error bars are the values of standard deviations.

confusion of the non-identical steps leads to the initial high error rates of GS. Additionally, the error rates of TE throughout the eight sessions fluctuate as the pressure-sensitive caret navigation has a lower bound of error rates, as shown in the prior studies [27, 55], and taking two consecutive caret navigation makes the error rates more random than only one caret navigation happened in the final step of copy and paste operations.

As for the final step of copy-and-paste operations (Section 3.5), two-way RM-ANOVA shows no effect of the interaction approaches on the error rate and the learning effect ($F_{1,7}$ = 1.61, p= 0.21) between sessions. Similarly, the caret navigation and swipes are both applied by GS(-CP, grey line) and TE(-CP, yellow line). As such, GS and TE result in similar average error rates of 6.04% (σ=0.02) and 6.10% (σ=0.02), respectively. Also, we highlight the initial high error rates (the first session) appeared in both the GS (mean=8.04%, σ=0.02) and TE (mean=8.39%, σ=0.03). The key reason is that the participants had mixed up the up-swipe and down-swipe gestures for the copy and paste operations. During the eighth session, the participants eventually reduce the error rates to 3.89% (σ=0.01) and 3.57% (σ=0.01) for GS and TE, which are highly comparable to the error rate of TE(-Text, orange line) solely supported by two caret navigation steps.

To sum up, TE is significantly faster and more accurate than GS. GS owns more lengthy procedures than TE, and thus are more error-prone and time-consuming. The majority of participants reported that GS is less straightforward than TE, as TE requires only visual searches of the positions for the two ends during the two caret navigation steps. In contrast, the scrolling gestures in GS demands continuous visual resources to keep track of the number of selected text units at the chosen granularity. For instance, the participants in Task 6 require to perform longish scrolling gestures at the character-level granularity for two sentences. The most extreme case can be the task of capturing the entire text minus one first character. Users with GS will perform tedious scrolling gestures at the character-level granularity for the whole text, while TE can simply obtain the two ends of the textual body. Additionally, the participants also reflected that TE was a more easy-to-apply solution than GS, because the three caret navigation steps (two in selecting texts and one in the subsequent copy-and-paste), as the backbone of TE, helps them to learn the new solution rapidly.

4.2 Interaction Area

Figure 16 shows the interaction footprint for TE (a) and GS (b) on a 375 pt * 668 pt touchscreen area captured per second. We first compute the interaction area by identifying the centre of the captured samples. Then, the interaction areas bounded by the black boxes (Horizontal (H) and Vertical (V)) are calculated by the three standard deviations from the centre (99.7% of the sample

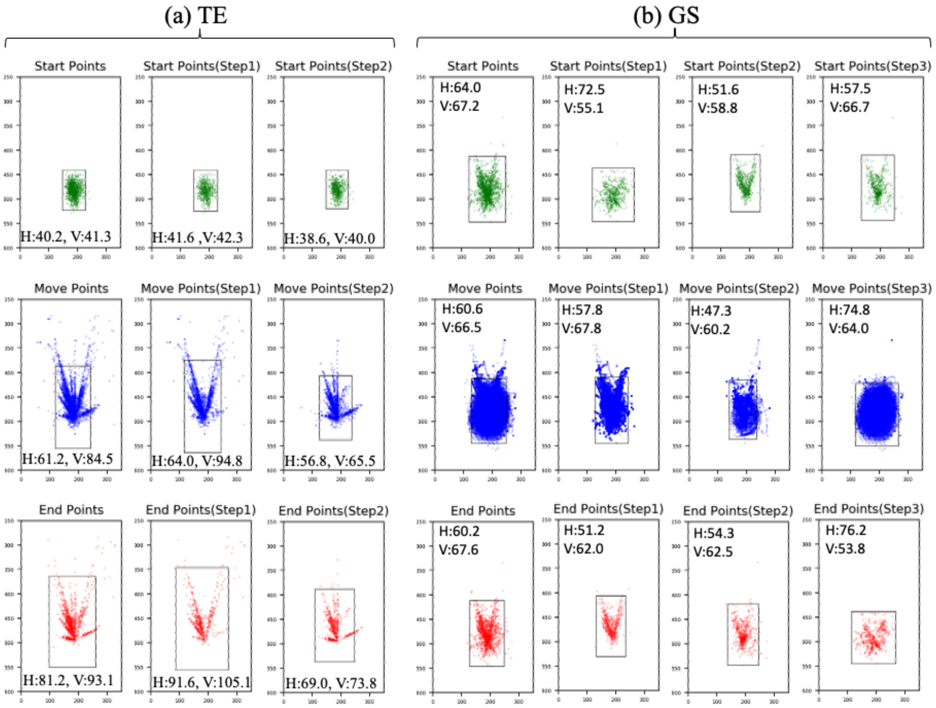


Fig. 16. Interaction footprint for TE (a) and GS (b), where dots in blue, green and red colours represent the overall move points, start points and end points; Black boxes show the central interaction areas; X and Y values are the horizontal and vertical dimensions of the interaction areas. The leftmost column of two solutions are the aggregated interaction footprint, the subsequent columns are the separated steps.

distribution). The leftmost column in TE and GS is the aggregated interaction footprint, and we take the maximum value of H and V among the start (green dots), move (blue dots), and end (red dots) points. We breakdown the footprints of respective steps in the following columns for better readability. TE and GS result in interaction areas of (81.2 pt * 93.1 pt) and 64.0 pt * 67.5 pt. The actual sizes of TE and GS are (9.96 mm * 10.51 mm) and (12.65 mm * 14.48 mm), respectively. The interaction area of GS (104.78 mm^2) is 42.80% smaller than TE (183.18 mm^2). We briefly explain the patterns in the interaction footprint (move points) in the respective steps of two solutions. TE demonstrates radial patterns in the first and second steps because they are solely supported by the caret navigation (Section 3.3) consisting of presses and subtle pan gestures. GS has three steps, and they are caret navigation (step 1), granularity management (step 2), and scrolling gestures (step 3). Step 1 of GS shows similar radial patterns, as shown in TE. Afterwards, the users press on a single point to finish the granularity selection, and hence the oval-shape footprint shows some random drops for the presses within the circular button. Finally, the scrolling gestures induce the user's thumb to act on the edge of the circular button, leading to the corresponding circular pattern in footprints. It is important to note that the final step of the copy and paste (Section 3.5 are mainly driven by the caret navigation and up and down swipes, which are similar to some procedures in TE (Step 1 and 2) and GS (Step 1). Therefore, the interaction footprint of copy and paste operations are not listed to avoid redundancy. To conclude, the caret navigation step is the most space-consuming gestures among all the proposed steps in the two solutions. Therefore, we

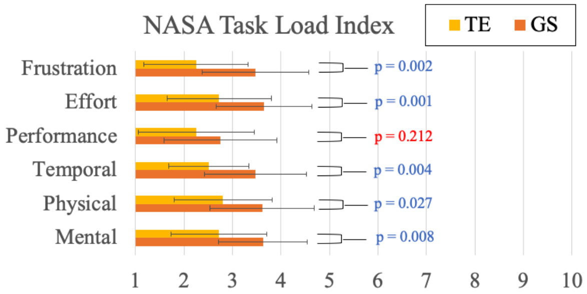


Fig. 17. NASA TLX results for GS and TE.

conclude that the proposed system (Press-n-Paste) requires a minimum area of (12.65 mm * 14.48 mm) to accommodate all the gestures.

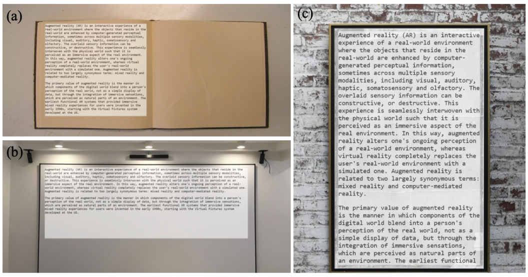
4.3 NASA TLX

Figure 17 displays the results of the NASA TLX survey. One-way ANOVA under two interaction approaches shows an effect of the interaction approach for all metrics (p <0.05) except for the Performance (p = 0.212). These results imply that the users perceive a difference between the two system solution in terms of Mental Demand, Physical Demand, Temporal Demand, Total Effort, and Frustration. In particular, the difference is most obvious in Mental, Temporal, Effort and Frustration (p <0.01). Participants reflect that GS demands a significantly higher temporal load (0.958 average point) than TE, and its usage leads to significantly higher efforts (0.931 average point) and frustration (1.22 average point).

5 EVALUATION II: USER ACCEPTANCE TO PRESS-N-PASTE

The first evaluation (Section 4) extensively discovers the user performance of pressure-sensitive copy-and-paste operations with indirect manipulations. In this section, the second evaluation aims to understand whether the users will have the attitude and intention to use the alternative forms of copy-and-paste on AR smartglasses. Instead of having prolonged usage with heavy-weighted smartglasses (i.e. Microsoft Hololens), we encourage the users to experience the entire cycle of PnP (TE) with fewer trials. Thus, we recruited another 37 participants (19 - 26 years old ($\overline{M}$ = 21.85, σ = 2.33) from a university campus. All of them are experience smartphone users but have no prior experience with pressure widgets. We conduct a 20-minute interview with each participant about the acceptance [23] to the new technology of PnP (TE). TE is chosen on the basis of its usability (i.e. completion time and accuracy). In the interview, the participants with TE run through three scenarios, including a museum, classroom and library, which are reasonable mock-up scenarios in our daily life, e.g. capturing text on the whiteboard inside the classroom. In each scenario, users with the AR headsets leveraging the image-to-text module (Section 3.6), as demonstrated in the experimental setting (Figure 9), choose 1 − 2 textual contents at random granularities. After completing the three scenarios, we explain the potential use cases of textual contents with AR smartglasses with the participants. The survey study was carried out complying the General Data Protection Regulation (GDPR) and approved by university institutional ethics review board (IRB) regulations. Figure 18 depicts the interfaces and contents of the three scenarios.

The participants fill in a questionnaire about the technology acceptance. The participants rate their technological literacy on a five-point Likert scale ranging from 1 to 5, 5 being the highest, with 1 being 'totally disagree' and 5 'totally agree'. The average technological literacy is very high, 3.6, ranging from 2 to 5, as copy-and-paste is their routine operations on smartphones. This survey



Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real-world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory. The overlaid sensory information can be constructive, or destructive. This experience is seamlessly interwoven with the physical world such that it is perceived as an immersive aspect of the real environment. In this way, augmented reality alters one's ongoing perception of a real-world environment, whereas virtual reality completely replaces the user's real-world environment with a simulated one. Augmented reality is related to two largely synonymous terms: mixed reality and computer-mediated reality.

The primary value of augmented reality is the manner in which components of the digital world blend into a person's perception of the real world, not as a simple display of data, but through the integration of immersive sensations, which are perceived as natural parts of an environment. The earliest functional

Fig. 18. The three daily scenario with textual contents: (a) library: Texts on a book; (b) classroom: Texts on a projector screen; and (c) museum: Texts on a description board for an artefact; the three scenarios employ a standard image-to-text module in OCR converting one text group from an image, as described in Section 3.6.

Table 2. Technology Acceptance Survey showing positive user feedback in three metrics of PU, PEOU and IU (Cronbach Alpha = 0.7859).

Survey Questions	MEAN	MED	MIN	MAX	STDEV
Perceived Usefulness (PU)					
TE would enable copy-and-paste conveniently.	3.25	3.75	1	5	1.070
TE would improve the way I deal for text in mobile AR.	3.2	4	1	5	1.005
TE is useful for the AR text CP.	3.40	3	1	5	0.940
TE is useful for mobile managing text and granularity.	3.15	3	1	5	0.988
Perceived Ease of Use (PEOU)					
Learning to use TE would be easy.	3.24	3	2	4	0.701
It would be easy for me to become skilful at using TE.	3.41	4	1	5	0.850
I find TE easy to use.	3.32	3	2	4	0.684
Intention Of Use (IOU)					
When TE is available, I intend to use it for dealing text CP.	3.76	4	3	5	0.598
When TE is available, I will use it with mobile headsets.	3.95	3.25	2	5	0.745
When TE is available, I would use it frequently.	3.49	3	2	5	0.608

aims at measuring three qualitative metrics: (1) Perceived Usefulness (PU); (2) Perceived Ease Of Use (PEOU); and (3) Intention Of Use (IOU). We compute the Cronbach Alpha reliability coefficients for the metrics. Among them, a coefficient above 0.70 denotes an acceptable consistency, as shown in Table 2.

In general, the participants are particularly positive to the three qualitative metrics. They found the TE to be easy to use (mean = 3.32, σ = 0.684), solving one of the major concern raised by the lack of copy-and-paste techniques on mobile headsets. Indeed, capturing textual contents without handling granularities was reported to be less efficient. Participants also considered the pressure-assisted copy-and-paste technique easy to learn (mean = 3.24, σ = 0.701) and to become skillful at (mean = 3.41, σ = 0.850). It is important to note that all the median values in the PEOU are no less than 3, which imply that the participants can pick up the TE effectively. The participants also considered the TE to be a convenient solution to the copy-and-paste operations (mean = 3.25, σ = 1.070). Although they considered that using the TE would improve the way they deal with textual contents (mean = 3.20, σ = 1.005), they found the prototype useful to augmented reality (AR) copy-and-paste (mean = 3.40, σ = 0.940), especially for managing various granularities in mobile scenario (mean = 3.15, σ = 0.988). The median values among the four questions are no less than 3, indicating that TE is perceived by the users positively. Finally, participants generally agree with using TE in the future, either in the daily commute, or as a complement to the mobile headsets. Users agree that they would use it frequently. Overall, the IOU is positive with TE, achieving our goal to make a convenient AR interface for subtle copy-and-paste in mobile scenarios. Although PU reflects higher variance values, the overall responses to TE across the three experimental scenarios are consistently positive.

6 DISCUSSION

Alternative modals: Prior work conducted studies of direct manipulation of texts on touch-screens [6], in which three interaction approaches for **the copy-and-paste (CP) operations** are compared. Two commercial solutions named Long-tap and 2-Fingers as well as a state-of-the-art solution named BezelCopy [6] result in completion times (per text selection) of 14,260 ms, 16,810 ms and 8,860 ms with accuracy rates of 99.3%, 97.45% and 96.29%, respectively. In contrast, GS and TE achieve average completion times of 193,074 ms and 163,148 ms (for 9 CP operations in each session), and average accuracy rates of 87.89% and 94.21% (taking the lower bounds in the entire text selection and CP procedures). These task completion times are equivalent to 21,453 ms and 18,128 ms per CP operation. In the final session, the task performance of GS and TE improves to 17,574 ms and 13,951 ms per CP operation, while the peak accuracy rates are 93.21% and 98.15%. Therefore, in comparison with the commercial solutions on touchscreen devices, Press-n-Paste (TE) makes a competitive performance time but a slightly lower accuracy rate, due to the inherited disadvantages from the transient property in pressure-sensitive interactions [8] and the indirect manipulations [52]. With more intensive training, the users with PnP (TE, 13,951 ms per CP) in the final session reach a practical range of CP operations, considering the reference performances from the commercial solutions on touchscreens, i.e. long-tap (14,260 ms per CP) and 2-Fingers (16,810 ms per CP) [6]. Furthermore, PnP enables users to manipulate the target texts with more direct solutions than voice-based methods. The major reasons are that the users with voice-based approaches are difficult to locate some specific characters and words at precise positions as well as various granularities [13]. Although voice-based approaches achieve a very efficient input bandwidth up to 152 word-per-minute [43], the time spent in voice-based manipulations within in-text environments such as word corrections and reviews is more than four times higher in the voice-based text inputs [3].

Design space of Pressure-sensitive CP operations: This paper first proposes system solutions for the entire cycle of copy-and-paste operations that covers the sequential procedures of selecting, copying and pasting texts. We acknowledge that the prior works on the design space of pressure widgets such as gauge modes with direct manipulations (7,100 ms per text selection), namely ForceSelect [10], menu selection (2,500 ms per selection in a 6-item numerical menu, with 83% accuracy rate) [50], and reaching various sized icons (60pt-icons: 1,554 ms per icon reach (97.22% accuracy rate) and 30pt-icons: 1,790 ms per icon reach (94.72% accuracy rate)) [7]. GS and TE are more lengthy procedures than the prior works, and they unavoidably result in longer average completion times of 21,453 ms and 18,128 ms per CP operation. We try to breakdown the time in each step on the basis of prior works, as follows. In TE, three caret navigation steps are involved. And we analog the time for triple icon reach (3 * 1,790 ms [7] = 5,370 ms). Thus, the surplus times (12,758 ms) are allocated to two presses and two swipes, in addition to some cognitive processing time. GS takes two caret navigation steps (one at the character selection and another at the paste operation) plus one menu selection. Equivalently, the time is 6,080 ms (2,500 ms [50] + 2 * 1,790 ms [7]). So the remaining time of 15,373 ms is taken by three presses, one scrolling gesture, and two swipes. According to the time difference in TE (15,373 ms) and GS (12,758 ms), we roughly estimate that the step of scrolling gesture in GS costs approximately at 2,615 ms.

In contrast, Section 4.1 presents the user performance from a different perspective, according to the recorded completion times for the individual steps in GS and TE. It can be said that the completion time taken in multi-step approaches allows the research community to re-validate the performance of pressure-sensitive interaction, and the highlights are as follows. First, our scenario employs the caret navigation for smaller targets (18-pt fonts) resulting in more challenging tasks and a hence lengthy target capturing process (4,411 ms, step 2 in TE). Next, our results reveal the difference in performance times (2,358 – 2,386 ms) for the identical steps of caret navigation (GS step 1; TE steps 1 and 2). The time gaps are regarded as the cognitive overheads for task planning. Accordingly, by comparing two closely related prior works [10, 50], a time gap of approximately 6,000 ms (per selection) exists between their studies, due to the complexity of task nature and corresponding planning efforts. In [50], menu items are sequentially listed in a menu with straightforward instructions of positioning the pointer in the correct sector, while the gauge mode [10] contains more varied tasks to select menu in multiple strategies through direct manipulations (pan gestures). In our multi-step approach (GS), we record an average time of 3,250 ms per menu selection during the second step of granularity management, where the cognitive overheads transfer to the previous step of caret navigation.

Furthermore, GS (87.89%) and TE (94.21%) show aligned accuracy rates compared with the prior works. TE relies on indirect reaches to small-size characters, so its accuracy rate is very similar to the prior works on selecting 30pt-icons (94.72%) [7]. Also, the bottleneck of GS happens at the step of granularity management (87.89% for 5-item menus) that is close to the 6-item menu with 83% accuracy, considering that increasing item numbers in a menu is inversely proportional to the selection accuracy [50].

Although the prior works indicate that human user can distinguish pressure up to multiple discrete levels when visual cues are available [12], very less knowledge is available to the design space of pressure widget for text-editing environments as well as copy-and-paste operations in particular. Our work sheds light to the designing pressure-sensitive widget for copy-and-paste operations, in which our evaluations show that the participants can leverage the pressure-sensitive caret navigation to indirectly manipulate the closely packed and small-size text objects. Among the two system solutions in PnP, we notice a trade-off between the interaction areas and user performance. That is, GS (104.78 mm^2) generates a significantly smaller interaction area than TE (183.18 mm^2). However, TE (163,148 ms and 5.79% error rate) outweighs GS (193,074 ms and 12.11%) in terms of performance time and accuracy. First, the difference in user performance is mainly caused by the step number and its natures in two solutions, in which TE has two identical steps and avoids the switch of tasks, but GS consists of three non-identical steps. Nevertheless, the two caret navigation steps in TE are more space-consuming than other steps of either pressing or scrolling gestures in GS that are easily bounded by the circular button. On the other hand, Press-n-Paste (PnP) not only serves as a proof of effective copy-and-paste operations on AR smartglasses, but also demonstrates the one-button interface for the potential uses on smartphones that keep growing in screen sizes, such as the foldable screens on Huawei Xmate 5G and Samsung Galaxy Fold. The one-handed thumb interaction, which is considered as a common scenario in mobile scenarios [27, 55], are impossible directly to reach all texts dispersed on these bigger and bigger screens [7], leading to a paradigm shift of copy-and-paste operations from direct to indirect manipulations.

Miniaturized interaction footprint: The interaction space we investigated can fit into the size-constrained smart wearables. A rising number of smart wearables are being launched on the market. Such wearables as smart rings and smart wristbands present a screen real estate even smaller than smartwatches. GS (9.96 mm * 10.51 mm) and TE (12.65 mm * 14.48 mm) can serve as promising solutions for copy-and-paste operations on these size-constrained wearables, where only a small (coin-sized) area of the pressure-sensitive touch interface is available. We also envision combined use of an MR headset and a finger-worn device [11] that is regarded as the fashionable wearables [45]. Moreover, it is feasible to put a pressure-sensitive button on the spectacle frame of the AR smartglasses. However, the pressure exertion on the spectacle frame may cause the sight movement of the smartglasses display, in case that the pressure-sensitive button locates at the vertical plane of the spectacle frame. Instead, we suggest that the button should locate at the horizontal plane and thus the users can employ the thumb and index finger to stabilize the operations on the spectacle frame. Additionally, flexible pressure sensors can be embedded in the textile as smart garments. In [27], a number of pressure-sensitive units (5 * 5 mm^2) are embedded in a smart glove, and the users can distinguish the character keys in the ambiguous keyboard within the finger space through employing the correct pressure levels.

Hybrid usage with other modalities: Our observations throughout the user evaluations indicate that the participants prefer to employ GS to handle lengthy texts. As textual contents can be managed with different granularities, we conjecture that a strategy of 'divide-and-conquer' can be considered to reduce the text length. Accordingly, an additional modality can be employed to divide the lengthy text into size no more than paragraph granularity. This encourages the users to employ a higher efficient approach of TE with divided text contents. For instance, voice commands allow users to select the target paragraph, e.g. the first paragraph, the second paragraph, and so on. Once the paragraph is selected, the user can apply TE to manage the textual contents quickly.

Application scenarios: The proposed solution can be implemented as an *interaction layer* [26] in web browsing and text editing environment on AR smartglasses, in response to the user's pain points, as described in Section 1. Also, the latest AR smartglasses connecting with smartphones[10,11] can also directly employ the proposed PnP solution. A real-life text-editing application supported by the proposed system solutions can be further developed to facilitate copy-and-paste operations with textual contents at various granularity in mobile scenarios. For example, a travelling app can potentially enable text translation of some selected texts on the AR mobile headsets. In the AR travelling app, the users can utilize the system cameras to capture the texts in the physical environment. The Optical Character Recognition (OCR) module turns the texts in the form of images to the text objects to be manipulated. Through selecting the interested text with PnP, the mobile headsets can process a lesser amount of the select text, instead of the entire text. The computationally constrained smartglasses gain benefits such as faster response time, less power consumption as well as saving network overheads [26]. One may argue how to manage the scenarios for the *'out-of-the-box'* texts. AR smartglasses suffer from the small screen real estate [28] that usually limits the user interaction inside a small field of view (FOV). Among all the smartglasses in the market, the users have to rotate their heads pointing to the region of interests in a *real-world surface* to receive the AR information in the limited FOV, instead of a fully immersive AR environment as depicted by marketing campaigns. In other words, all the user interaction with texts (and even images/graphics) are bounded by a small rectangle shaped FOV, and hence the major design issue comes to the image-to-text management. When the user is going to take an image through the smartglasses camera, it is suggested that the user's head should point to the region of interest, i.e. the textual cluster. Next, the OCR only transcribes the pointed text cluster, and another full-page menu shows the texts for user interaction with the proposed PnP approach.

Towards Ring-form interfaces for highly mobile AR interaction. Rather than employing an opportunistic approach to build the smart ring surface, it is necessary to acquire the knowledge about the required interaction area on a smart ring, especially when pressure is considered as an input modality. A work leveraging pressure-sensitive interaction on a circle button for drone flights also demonstrates a miniaturized area of 10.8 mm * 6.1 mm [53], where a drone can offer an extended view in AR and serves as a remarkable example of IoT-AR interaction. Under the limited surface of a smart ring, it is better to enrich the function of a smart ring as many as possible, including interacting with digital contents in AR and controlling IoT objects with AR smartglasses.

[10] Samsung Nreal: https://nreal.ai

[11] Mad Gaze Glow: https://www.madgaze.com/glow/

Considering our studies together with [53], this work further contributes to the knowledge of minimal interaction areas on a ring-form surface. Thus, with TE, an area of 12.65 mm * 14.48 mm is sufficient for copy-and-paste operations (the entire solution of PnP), targeting small and dense items (GS and TE), as well as controlling drone flights with four pairs of directional movements. In other words, AR smartglasses users with such minimal interfaces on a smart ring can interact with textual contents either in the physical surroundings or in a distant location through the drone telepresence.

7 CONCLUSION

In this paper, we designed Press-n-Paste (PnP) for the copy-and-paste (CP) operations through indirect manipulation on AR smartglasses. In PnP, two system solutions named Granularity Scrolling (GS) and Two Ends (TE) leverage touch-sensitive and pressure-sensitive surface to achieve flexible CP at various granularities. Our evaluation shows that both GS and TE are highly usable. After eight training sessions, our participants with GS and TE reach the peak performance of 17,574 ms and 13,951 ms per CP operation, with 93.21% and 98.15% accuracy rates respectively, which are comparable to the commercial standards using direct manipulation on touchscreens. Our participants reflect reasonable workloads in NASA TLX questionnaires, and positive rating of user acceptance in terms of PU, PEOU and IU. Additionally, our participants with PnP complete the copy-and-paste operations within a miniaturized footprint no bigger than 183.172 mm^2 (12.65 mm * 14.48 mm). The measured footprint serves as a quantitative proof not only to the multi-step CP operations but also some simpler tasks such as interacting with larger icons and menus. Without scarifying the usability, smart wearables can include such an area in their design for the sake the mobility. Also, PnP broadens the design space of pressure-sensitive widgets for multi-step interaction tasks. For future work, we plan to apply the solutions into some real-life augmented reality applications. Also, we will leverage the results, especially the interaction footprint, to develop a finger-worn device for user interaction with AR smartglasses. Moreover, we will re-run the user study with the senior citizens to further validate the robustness of our solutions, in case that the older adults are a less proficient group to pressure widgets due to weaker thumbs. Also, we will investigate how GS can be employed in other languages, for example, the Chinese characters and words are ambiguous, so we have to design new strategies of granularity management considering Natural Language Processing (NLP).

REFERENCES

[1] Toshiyuki Ando, Toshiya Isomoto, Buntarou Shizuki, and Shin Takahashi. 2018. Press Tilt: One-Handed Text Selection and Command Execution on Smartphone. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction (OzCHI '18)*. Association for Computing Machinery, New York, NY, USA, 401–405. https://doi.org/10.1145/3292147.3292178

[2] Axel Antoine, Sylvain Malacria, and Géry Casiez. 2017. ForceEdge: Controlling Autoscroll on Both Desktop and Mobile Computers Using the Force. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3281–3292. https://doi.org/10.1145/3025453.3025605

[3] Shiri Azenkot and Nicole B. Lee. 2013. Exploring the Use of Speech Input by Blind People on Mobile Devices. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, Article Article 11, 8 pages. https://doi.org/10.1145/2513383.2513440

[4] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1393–1402. https://doi.org/10.1145/2702123.2702252

[5] Olivier Chapuis and Nicolas Roussel. 2007. Copy-and-Paste between Overlapping Windows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 201–210. https://doi.org/10.1145/1240624.1240657

[6] Chen Chen, Simon T. Perrault, Shengdong Zhao, and Wei Tsang Ooi. 2014. BezelCopy: An Efficient Cross-Application Copy-Paste Technique for Touchscreen Smartphones. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. Association for Computing Machinery, New York, NY, USA, 185–192. https://doi.org/10.1145/2598153.2598162

[7] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 212, 12 pages. https://doi.org/10.1145/3290605.3300442

[8] Christian Corsten, Simon Voelker, and Jan Borchers. 2017. Release, Don't Wait! Reliable Force Input Confirmation with Quick Release. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. Association for Computing Machinery, New York, NY, USA, 246–251. https://doi.org/10.1145/3132272.3134116

[9] Vittorio Fuccella, Poika Isokoski, and Benoit Martin. 2013. Gestures and Widgets: Performance in Text Editing on Multi-Touch Capable Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2785–2794. https://doi.org/10.1145/2470654.2481385

[10] Alix Goguey, Sylvain Malacria, and Carl Gutwin. 2018. Improving Discoverability and Expert Performance in Force-Sensitive Text Selection for Touch Devices with Mode Gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 477, 12 pages. https://doi.org/10.1145/3173574.3174051

[11] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. https://doi.org/10.1145/3332165.3347947

[12] Seongkook Heo and Geehyuk Lee. 2011. Force Gestures: Augmenting Touch Screen Gestures with Normal and Tangential Forces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 621–626. https://doi.org/10.1145/2047196.2047278

[13] Jonggi Hong and Leah Findlater. 2018. Identifying Speech Input Errors Through Audio-Only Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 567, 12 pages. https://doi.org/10.1145/3173574.3174141

[14] Eva Hornecker. 2012. Beyond Affordance: Tangibles' Hybrid Nature. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. Association for Computing Machinery, New York, NY, USA, 175–182. https://doi.org/10.1145/2148131.2148168

[15] Weiyang Huan, Huawei Tu, and Zhuying Li. 2019. Enabling Finger Pointing Based Text Selection on Touchscreen Mobile Devices. In *Proceedings of the Seventh International Symposium of Chinese CHI (Chinese CHI '19)*. Association for Computing Machinery, New York, NY, USA, 93–96. https://doi.org/10.1145/3332169.3332172

[16] NASA AMES Research Center Human Performance Research Group. 1999. *NASA Task Load Index (TLX)*. https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX.pdf

[17] Kaori Ikematsu and Itiro Siio. 2013. Memory Stones: An Intuitive Copy-and-Paste Method between Multi-Touch Computers. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. Association for Computing Machinery, New York, NY, USA, 1287–1292. https://doi.org/10.1145/2468356.2468586

[18] Howell Istance, Richard Bates, Aulikki Hyrskykari, and Stephen Vickers. 2008. Snap Clutch, a Moded Approach to Solving the Midas Touch Problem. *Eye Tracking Research and Applications Symposium (ETRA)*, 221–228. https://doi.org/10.1145/1344471.1344523

[19] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2291–2300. https://doi.org/10.1145/2470654.2481317

[20] R. Kishi and T. Hayashi. 2015. Effective gazewriting with support of text copy and paste. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*. 125–130. https://doi.org/10.1109/ICIS.2015.7166581

[21] Abinaya Kumar, Aishwarya Radjesh, Sven Mayer, and Huy Viet Le. 2019. Improving the Input Accuracy of Touchscreens Using Deep Learning. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper LBW1514, 6 pages. https://doi.org/10.1145/3290607.3312928

[22] Jianwei Lai, Navid Rajabi, and Elahe Javadi. 2019. A Shortcut for Caret Positioning on Touch-Screen Phones. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 35, 6 pages. https://doi.org/10.1145/3338286.3340146

[23] K. Y. Lam, L. H. Lee, T. Braud, and P. Hui. 2019. M2A: A Framework for Visualizing Information from Mobile Web to Mobile Augmented Reality. In *2019 IEEE International Conference on Pervasive Computing and Communications*

(PerCom. 1–10. https://doi.org/10.1109/PERCOM.2019.8767388

[24] Felix Lauber, Anna Follmann, and Andreas Butz. 2014. What You See is What You Touch: Visualizing Touch Screen Interaction in the Head-up Display. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. Association for Computing Machinery, New York, NY, USA, 171–180. https://doi.org/10.1145/2598510.2598521

[25] L. Lee and P. Hui. 2018. Interaction Methods for Smart Glasses: A Survey. *IEEE Access* 6 (2018), 28712–28732. https://doi.org/10.1109/ACCESS.2018.2831081

[26] Lik Hang Lee, Tristan Braud, Farshid Hassani Bijarbooneh, and Pan Hui. 2019. TiPoint: Detecting Fingertip for Mid-Air Interaction on Computational Resource Constrained Smartglasses. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. Association for Computing Machinery, New York, NY, USA, 118–122. https://doi.org/10.1145/3341163.3347723

[27] Lik Hang Lee, Kit Yung Lam, Tong Li, Tristan Braud, Xiang Su, and Pan Hui. 2019. Quadmetric Optimized Thumb-to-Finger Interaction for Force Assisted One-Handed Text Entry on Mobile Headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article Article 94 (Sept. 2019), 27 pages. https://doi.org/10.1145/3351252

[28] L. H. Lee, K. Yung Lam, Y. P. Yau, T. Braud, and P. Hui. 2019. HIBEY: Hide the Keyboard in Augmented Reality. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom.* 1–10. https://doi.org/10.1109/PERCOM.2019.8767420

[29] Hannah Limerick, James W. Moore, and David Coyle. 2015. Empirical Evidence for a Diminished Sense of Agency in Speech Interfaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3967–3970. https://doi.org/10.1145/2702123.2702379

[30] Sylvain Malacria, Jonathan Aceituno, Philip Quinn, Géry Casiez, Andy Cockburn, and Nicolas Roussel. 2015. Push-Edge and Slide-Edge: Scrolling by Pushing Against the Viewport Edge. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2773–2776. https://doi.org/10.1145/2702123.2702132

[31] Yuki Matsuura, Tsutomu Terada, Tomohiro Aoki, Susumu Sonoda, Naoya Isoyama, and Masahiko Tsukamoto. 2019. Readability and Legibility of Fonts Considering Shakiness of Head Mounted Displays. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. Association for Computing Machinery, New York, NY, USA, 150–159. https://doi.org/10.1145/3341163.3347748

[32] Motoki Miura and Kenji Saisho. 2014. A Text Selection Technique Using Word Snapping. *Procedia Computer Science* 35 (2014), 1644 – 1651. https://doi.org/10.1016/j.procs.2014.08.257 Knowledge-Based and Intelligent Information Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

[33] Sachi Mizobuchi, Shinya Terasaki, Turo Keski-Jaskari, Jari Nousiainen, Matti Ryynanen, and Miika Silfverberg. 2005. Making an Impression: Force-Controlled Pen Input for Handheld Devices. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1661–1664. https://doi.org/10.1145/1056808.1056991

[34] P. Mohan, W. B. Goh, C. Fu, and S. Yeung. 2018. DualGaze: Addressing the Midas Touch Problem in Gaze Mediated VR Interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 79–84.

[35] Alex Olwal, Steven Feiner, and Susanna Heyman. 2008. Rubbing and Tapping for Precise and Rapid Selection on Touch-Screen Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 295–304. https://doi.org/10.1145/1357054.1357105

[36] Henri Palleis, Julie Wagner, and Heinrich Hussmann. 2016. Novel Indirect Touch Input Techniques Applied to Finger-Forming 3D Models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16)*. Association for Computing Machinery, New York, NY, USA, 228–235. https://doi.org/10.1145/2909132.2909257

[37] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand. 2019. BASNet: Boundary-Aware Salient Object Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7471–7481.

[38] Gonzalo Ramos and Ravin Balakrishnan. 2005. Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. Association for Computing Machinery, New York, NY, USA, 143–152. https://doi.org/10.1145/1095034.1095059

[39] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure Widgets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. Association for Computing Machinery, New York, NY, USA, 487–494. https://doi.org/10.1145/985692.985754

[40] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure Widgets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. Association for Computing Machinery, New York, NY, USA, 487–494. https://doi.org/10.1145/985692.985754

[41] Umar Rashid, Miguel A. Nacenta, and Aaron Quigley. 2012. Factors Influencing Visual Attention Switch in Multi-Display User Interfaces: A Survey. In *Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis '12)*. Association for Computing Machinery, New York, NY, USA, Article Article 1, 6 pages. https://doi.org/10.1145/2307798.2307799

[42] Radiah Rivu, Yasmeen Abdrabou, Ken Pfeuffer, Mariam Hassib, and Florian Alt. 2020. Gaze'N'Touch: Enhancing Text Selection on Mobile Devices Using Gaze. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382802

[43] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article Article 159 (Jan. 2018), 23 pages. https://doi.org/10.1145/3161187

[44] Dominik Schmidt, Corina Sas, and Hans Gellersen. 2013. Personal Clipboards for Individual Copy-and-Paste on Shared Multi-User Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 3335–3344. https://doi.org/10.1145/2470654.2466457

[45] Roy Shilkrot, Jochen Huber, Jürgen Steimle, Suranga Nanayakkara, and Pattie Maes. 2015. Digital Digits: A Comprehensive Survey of Finger Augmentation Devices. *ACM Comput. Surv.* 48, 2, Article Article 30 (Nov. 2015), 29 pages. https://doi.org/10.1145/2828993

[46] Jeffrey Stylos, Brad A. Myers, and Andrew Faulring. 2004. Citrine: Providing Intelligent Copy-and-Paste. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. Association for Computing Machinery, New York, NY, USA, 185–188. https://doi.org/10.1145/1029632.1029665

[47] Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. 2016. Fix and Slide: Caret Navigation with Movable Background. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 478–482. https://doi.org/10.1145/2935334.2935357

[48] Kenji Suzuki, Ryuuki Sakamoto, Daisuke Sakamoto, and Tetsuo Ono. 2018. Pressure-Sensitive Zooming-out Interfaces for One-Handed Mobile Interaction. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, Article Article 30, 8 pages. https://doi.org/10.1145/3229434.3229446

[49] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article Article 113 (Sept. 2017), 21 pages. https://doi.org/10.1145/3130978

[50] Graham Wilson, Craig Stewart, and Stephen A. Brewster. 2010. Pressure-Based Menu Selection for Mobile Devices. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/1851600.1851631

[51] R. Xiao, J. Schwarz, N. Throm, A. D. Wilson, and H. Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660.

[52] Zhican Yang, Chun Yu, Xin Yi, and Yuanchun Shi. 2019. Investigating Gesture Typing for Indirect Touch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article Article 117 (Sept. 2019), 22 pages. https://doi.org/10.1145/3351275

[53] Yui-Pan Yau, Lik Hang Lee, Zheng Li, Tristan Braud, Yi-Hsuan Ho, and Pan Hui. 2020. How Subtle Can It Get? A Trimodal Study of Ring-Sized Interfaces for One-Handed Drone Control. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 63 (June 2020), 29 pages. https://doi.org/10.1145/3397319

[54] Shengdong Zhao, Fanny Chevalier, Wei Tsang Ooi, Chee Yuan Lee, and Arpit Agarwal. 2012. AutoComPaste: Auto-Completing Text as an Alternative to Copy-Paste. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. Association for Computing Machinery, New York, NY, USA, 365–372. https://doi.org/10.1145/2254556.2254626

[55] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu, and Yuanchun Shi. 2018. ForceBoard: Subtle Text Entry Leveraging Pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 528, 10 pages. https://doi.org/10.1145/3173574.3174102

[56] Fengyuan Zhu and Tovi Grossman. 2020. BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376233

Received April 2020; revised July 2020; accepted October 2020