M[eye]cro : Eye-gaze+Microgestures for Multitasking andInterruptions

Jérémy Wambecke, Alix Goguey, Laurence Nigay, Lauren Dargent, Daniel Hauret, Stéphanie Lafon, Jean-Samuel Louis de Visme

M[eye]cro: Eye-gaze+Microgestures for Multitasking and Interruptions

JÉRÉMY WAMBECKE, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France
ALIX GOGUEY, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France
LAURENCE NIGAY, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France
LAUREN DARGENT, Dassault Aviation, Saint-Cloud, France
DANIEL HAURET, Flight Avionics / Thales Avionics, Mérignac, France
STÉPHANIE LAFON, Flight Avionics / Thales Avionics, Mérignac, France
JEAN-SAMUEL LOUIS DE VISME, Flight Avionics / Thales Avionics, Mérignac, France

We present M[eye]cro an interaction technique to select on-screen objects and navigate menus through the synergistic use of eye-gaze and thumb-to-finger microgestures. Thumb-to-finger microgestures are gestures performed with the thumb of a hand onto the fingers of the same hand. The active body of research on microgestures highlights expected properties including speed, availability and eye-free interaction. Such properties make microgestures a good candidate for multitasking. However, while praised, the state-of-the-art hypothesis stating that microgestures could be beneficial for multitasking has never been quantitatively verified. We study and compare M[eye]cro to a baseline, i.e., a technique based on physical controllers, in a cockpit-based context. This context allows us to design a controlled experiment involving multitasking with low- and high-priority tasks in parallel. Our results show that performances of the two techniques are similar when participants only perform the selection task. However, M[eye]cro tends to yield better time performance when participants additionally need to treat high-priority tasks in parallel. Results also show that M[eye]cro induces less fatigue and is mostly preferred.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**.

Additional Key Words and Phrases: Input Techniques; Touch; Haptic; Pointing; Gesture; Transportation

ACM Reference Format:
Jérémy Wambecke, Alix Goguey, Laurence Nigay, Lauren Dargent, Daniel Hauret, Stéphanie Lafon, and Jean-Samuel Louis de Visme. 2021. M[eye]cro: Eye-gaze+Microgestures for Multitasking and Interruptions. *Proc. ACM Hum.-Comput. Interact.* 5, EICS, Article 210 (June 2021), 22 pages. https://doi.org/10.1145/3461732

1 INTRODUCTION

Thumb-to-finger gestures, i.e., single hand microgestures, are gestures made with the thumb on other fingers. Thumb-to-finger interaction offers an expressive set of gestures, e.g., taps with variable pressure, swipes, shape drawing and time-based gestures such as dwell taps or double taps. They can be performed on several parts of the fingers and take advantage of the proprioception

Authors' addresses: Jérémy Wambecke, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France; Alix Goguey, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France; Laurence Nigay, Laboratoire Informatique de Grenoble, Université Grenoble Alpes, France; Lauren Dargent, Dassault Aviation, Saint-Cloud, France; Daniel Hauret, Flight Avionics / Thales Avionics, Mérignac, France; Stéphanie Lafon, Flight Avionics / Thales Avionics, Mérignac, France; Jean-Samuel Louis de Visme, Flight Avionics / Thales Avionics, Mérignac, France.

2573-0142/2021/6-ART210 $15.00
https://doi.org/10.1145/3461732

and hand dexterity. Previous work on thumb-to-finger gestures has mainly focused on gesture recognition, rather than interaction. As an interaction modality, thumb-to-finger gestures could potentially: be performed eyes-free (capitalizing on human proprioception) and thus while the visual attention is focused on another parallel task; be performed *in-situ* allowing them to be interrupted so as to quickly switch to any other interactors; and be performed swiftly. Thumb-to-finger gestures therefore define a promising interaction modality for multitasking, when users need to quickly perform a task while doing another one, e.g., changing the volume of the radio while driving or modifying an object on a map while piloting an aircraft. Moreover, a huge benefit of microgestures is their potential to be performed with one hand while the other hand performs another task. Such benefits are expected and illustrated in cycling and driving contexts [19, 43, 54, 54]. However, in these studies the benefits of thumb-to-finger microgestures are not quantitatively assessed.

In this paper, we present the first quantitative study of multitasking involving thumb-to-finger gestures. We study multitasking and task switching by considering both parallel tasks and task interruption (alert management). The high-priority task is performed using the right hand. Microgestures performed with the left hand are readily available at all times and used to perform the low-priority task. This hand mapping is directly inspired from cockpit layouts. However, in our experiments we ensured that all participants were right-handed to avoid use of the non-dominant hand for high-priority tasks. In this study, for performing the low-priority task, we compare a new thumb-to-finger gesture based technique, M[EYE]CRO, with a technique based on physical controllers. M[EYE]CRO is a technique to select on-screen objects using eye-gaze and navigate menus using thumb-to-finger gestures. In an object selection phase, M[EYE]CRO takes advantage of eye movements to quickly point at targets, combined with thumb-to-finger microgestures to avoid the Midas touch problem. In a menu, M[EYE]CRO uses thumb-to-finger microgestures to select items without requiring users to grasp a specific device.

We choose an aircraft cockpit as a usage context for studying multitasking in a critical environment. It is now common for pilots to spend most of their time in the cockpit monitoring several on-board instruments (e.g., airspeed indicator and compass) or managing the remaining route. They also select different objects on the on-board screens to tweak or read their associated values (e.g., objects on a map). Depending on the flight phase, these tasks can even occur in parallel. This context is thus proper for studying multitasking involving thumb-to-finger microgestures since we can define a pseudo-realistic task in which microgestures are performed in parallel to piloting and can be interrupted by external stimuli. The study protocol was designed jointly with aviation professionals (i.e., experienced pilots, aviation ergonomists and cockpit designers). Their expertise helped ensure a realistic task scenario as well as a realistic baseline technique based on physical controllers (e.g., physical buttons and knobs). We compared M[EYE]CRO to an on-board aircraft tangible-based interaction technique in a controlled experiment looking at three different contexts: 1/ users only performing a low-priority selection task (which serves as our baseline comparison between both techniques); 2/ while performing the low-priority task, users dealing with a high-priority intermittent alert monitoring task (to evaluate selection performances while users' attention is divided, and need to abruptly interrupt their current task); and 3/ while performing the low-priority task, users performing a high-priority continuous control task (to evaluate selection performances while another task is performed in parallel).

Our results show that performances of the two techniques are similar when participants only perform a selection task. However, M[EYE]CRO tends to yield better time performances when other tasks are involved. M[EYE]CRO also induces less fatigue and is mostly preferred by the participants. From these results, we can conclude that the microgesture-based technique tends to be similar in performance to the technique based on physical controllers when no additional task is conducted. However, the microgesture-based technique seems to show better performance than the technique based on physical controllers when additional tasks are performed in parallel. This is very promising for microgesture-based techniques as the results quantitatively confirm a state-of-the-art hypothesis. Finally, although we tested M[EYE]CRO in a very specific context, this technique for object and command selection could easily be adapted to several contexts of use. But further studies would nonetheless be needed to confirm M[EYE]CRO's benefits.

This paper makes two main contributions. 1/ We present M[EYE]CRO, an interaction technique using eye-gaze and thumb-to-finger microgestures for object and command selection, and compare it to a technique based on physical controllers in a cockpit context. 2/ We conduct a study to quantitatively assess the benefits of thumb-to-finger microgestures for multitasking in a critical context of use, in which microgestures are performed in parallel to another task and are interrupted.

2 RELATED WORK

We build on previous work on microgestures and gaze selection, the two modalities of the M[EYE]CRO technique that are used in a synergistic way. We also present related work on interaction in the cockpit, which is the studied application domain of M[EYE]CRO.

2.1 Microgestures and Thumb-to-finger Gestures

Microgestures are defined as gestures performed with the thumb of a hand onto the fingers of the same hand [7]. Several elicitation studies have been conducted to define appropriate microgestures, either hand free [7] or while grasping objects [6, 39, 50, 51]. Thumb-to-finger gestures are microgestures involving a contact between the thumb and another hand part [41]. Several studies assess the comfort of finger parts on which to perform thumb-to-finger gestures [18, 21, 36, 46]. Results show that comfort gradually decreases from the index finger to the pinky, however, the index and middle fingers are close in terms of comfort compared to the other fingers. Comfort also tends to decrease along each finger from the top phalanx to the bottom one. Kao *et al.* showed that performing gestures on the nail is suited for interaction and appreciated by users [23].

To recognize thumb-to-finger gestures several approaches use cameras, however, they require calibration and they suffer from occlusion (i.e occlusion of joints and finger segments) [29, 36, 41]. Other approaches equip the hand with touch sensors, which seem to be a more reliable and easy to design approach [21, 28, 45, 52, 56, 57]. This is the chosen method for our technique M[EYE]CRO.

Beyond microgesture elicitation and recognition, few studies focused on interaction techniques based on microgestures. Tulip Menu was designed to select menu items using finger pinches [5] in virtual/augmented reality. Chan *et al.* designed a trackpad onto the tip of the index finger to select targets [8]. However, this technique was not quantitatively compared to another technique. The other techniques were designed for text entry using thumb-to-finger gestures [21, 28, 49, 52, 55]. All these techniques were compared to standard keyboards, but most importantly none were tested in a multitasking context.

The use of microgestures in cycling and driving contexts was also studied. Tan *et al.* found no performance difference between microgesture interaction and physical controller interaction while cycling [43]. Also in a cycling context, Xiao *et al.* observed that microgestures did not impact reaction time, however, they impacted the detection of objects in the peripheral vision [53]. Two studies in a driving context applied microgestures to control car functions [19, 54]. The first study found no difference on driving between microgestures and free hand gestures [19]. The second study showed that the reaction time was not impacted by the use of microgestures [54]. These contextual studies showed that microgestures are promising for performing low-priority tasks. However, all these studies were performed with the hand grabbing the steering wheel without considering thumb-to-finger gestures. In the cockpit context of our study the hand can be released to perform thumb-to-finger gestures allowing a much richer vocabulary of microgestures.

2.2 Gaze Selection

Pointing using eye-gaze has been extensively studied in the literature. A set of techniques support classical cursor selection with eye-gaze [3, 15, 24, 58]. Other techniques are dedicated to text typing with eye-gaze [1, 17, 26]. We focus on target selection with eye-gaze. Sibert *et al.* compared eye-gaze selection, using a dwell time for target validation, with the mouse [40]. They observed that eye-gaze leads to better performances than the mouse on sufficiently spaced targets. Kumar *et al.* showed that a technique based on eye-gaze and keyboard validation has performances close to using a mouse for a web browsing task [25]. The results of an experiment conducted by Zhang *et al.* showed that validation with a keyboard press tends to be faster than using a dwell time [59].

Eye-gaze selection is promising for target selection if using a classical mouse cursor is not possible. For instance, it is commonly used as a selection mean for augmented or virtual reality [4, 27, 37]. For interaction with a head-mounted display, Jalaliniya *et al.* observed that eye-gaze pointing is faster than pointing with a hand-held finger mouse used as a baseline [20]. Luro *et al.* compared eye-gaze selection with a traditional hand controller in virtual reality. Both methods led to similar selection time but the traditional hand controller was more accurate, while eye-gaze produced less perceived cognitive load [30]. In GazeGrip [60], Zhou *et al.* combined eye-gaze with back-of-device touch interaction. Preliminary results suggest that the system reduced fatigue and improve accuracy and precision.

Combining touch and eye-gaze was studied as a promising way to design interaction techniques. Voelker *et al.* studied the combined usage of touch and eye-gaze in an interactive workspace consisting of an horizontal touchscreen and a vertical touchscreen. Voelker *et al.* observed that using gaze input and horizontal touchscreen input to validate the selection of a target displayed on a vertical touchscreen is faster than directly touching the target on the vertical touchscreen [47]. Pfeuffer *et al.* compared touch+gaze with touch only, on rotation, scaling and dragging tasks [33]. Results showed that gaze+touch is as fast and accurate as touch only for rotation and scaling but is slower and less precise for dragging. A technique using eye-gaze and index-middle finger pinch gestures was also designed but was not compared to other techniques [34]. An informal evaluation nonetheless showed an efficient adoption of this technique by users who found it natural.

Finally, Prabhakar *et al.* [35] compared the usage of eye-gaze with a touchscreen to interact while driving a car. Results showed that eye-gaze leads to better driving performances than when using a touchscreen, which is promising for the use of eye-gaze when performing tasks in parallel.

2.3 Interaction in the Cockpit

Most of the studies on interaction in the cockpit focused on physical controllers and touchscreens. Stanton *et al.* compared four modalities (i.e., trackpad, trackball, rotary controllers and touchscreen) for performing two tasks (i.e., target selection and menu navigation) [42]. They found that touchscreen interaction seems to be the fastest modality. However, this modality was ranked higher in terms of discomfort than the other modalities. Rogers *et al.* compared a touchscreen interaction to rotary knobs to control multiple widgets (e.g., buttons, sliders) [38]. Results showed that no modality performs better overall and that performance is dependent on the task at hand and users' demographics. Voelker *et al.* performed an experiment to compare virtual rotary knobs on a touchscreen with physical ones for a simple rotation task [48]. Results showed that physical knobs are 20% faster, and their performance is barely impacted when not looking at them, in contrast with touchscreen interaction which requires considerably more visual attention. Thomas compared physical cursor control devices (i.e., thumbsticks and a trackball) with a trackpad and a touchscreen on a 2D target selection task [44]. The trackpad tends to be faster and more accurate, followed by the touch modality and then the physical controllers. However, the touch modality performed worst for small target selections. Alapetite *et al.* compared classical cockpit physical controllers with the touch modality in a real-world context: the modification of a flight planning in a cockpit simulator [2]. The main task was performed in parallel with an alert monitoring task. Pilots performing the study needed to take actions when alerts went off. They observed that pilots were faster with physical controllers than with touchscreen interaction. In all these studies, physical controllers seem to perform better than touch interaction. It is mainly due to the high visual attention that is required for touch interaction [48]. Moreover, several other studies indicate that the performance of touch interaction decreases considerably with degraded conditions (e.g., operator stress, higher cognitive load, smoke or turbulence) [12, 13, 22].

Turbulent Touch uses a stencil to stabilize touch interaction during vibrations [9]. Compared to classical touchscreen inputs and a trackball, Turbulent Touch did not perform faster but was less error prone for small targets in a high vibration context. Similarly Braced Touch uses multiple fingers as anchors on the screen to stabilize touch interaction [10]. Results showed that using multiple fingers instead of only one leads to better performances in a vibration condition. Both studies were performed on a vibrating platform and asked participants to manage alerts during the tasks. GazeForm [31] and Multi-Plié [32] are two interfaces combining touch and physical interaction. GazeForm is a classical touchscreen which turns into a physical interface (i.e., a rotary encoder emerges from the surface) when the user does not look at it anymore. A comparison with classical touch interaction showed a gain of 20% in execution time. Multi-Plié is an accordion touchscreen whose shape can be changed by the pilot or automatically depending on the context. Two prototypes of Multi-Plié have been designed after gathering requirements from professional pilots.

Most of the studies improving interaction in the cockpit have been centered on touch interaction and on how to improve its use in a cockpit, especially in a vibrating environment. This focus on touch interaction is explained by the desired goal of the aviation industry to use modalities that are more flexible than physical controllers. Adding or modifying physical controllers is at best very costly for manufacturers, at worst impossible. Therefore, microgestures are appropriate in this context since multiple interaction tasks can be mapped on the same set of gestures. However, to our knowledge, no study of gestures (especially microgestures) with eye-tracking has been conducted either on their usage in the cockpit or on their comparison with physical controllers.

3 INTERFACE AND TECHNIQUES

In this section we describe the interface (both software and hardware) that was used in the experiment. We first describe the low-priority task and how one can complete it using both tested techniques (i.e., BASELINE and M[EYE]CRO). We then describe the high-priority tasks that were used in some conditions of the experiment and how to manage them. As stated in the introduction, aviation professionals (i.e., experienced pilots, cockpit designers and aviation ergonomists) were involved in the design of the interface, the techniques and the tasks. In particular the resulting studied tasks resemble tasks commonly performed in a cockpit while being abstract enough to be performed by non-expert users. Abstracting tasks also makes it possible to anticipate a generalization of the results, although more studies will be required.

3.1 Low-Priority Task

Our goal with the low-priority task was to approach a realistic sequence of actions that occurs in a cockpit and involves both phases of object and property selection. With aviation professionals, we chose to base our low-priority task on a widespread flying routine in which pilots select objects on a map and apply commands chosen from a contextual menu. We used an abstract version of this task.

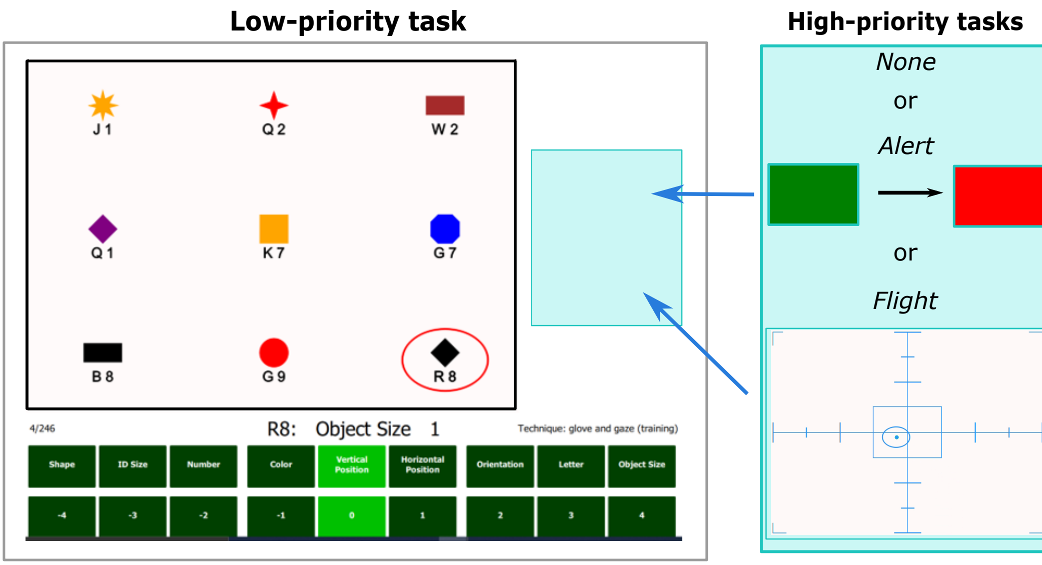

Fig. 1. Illustration of the low-priority task and high-priority tasks used in the experiment. The left image corresponds to the screen seen by the participants during the experiment. The top left rectangle with black outline displays a 3 by 3 grid of objects that was used for the object selection phase. The menu used to modify the object properties lies at the bottom of this grid. Between these two parts the instruction of the task is displayed: in the example, the participant has to select the object R8 (selection already done in this example) and to modify its `object size` to 1. The right part of the interface (blue on the illustration) is dedicated to the high-priority task if any. It is left empty in the NoSecondary condition, filled by a green rectangle which becomes red when an alert goes off in the Alert condition, or with a flight instrument in the Flight condition.

The low-priority task is composed of two phases: the object selection phase and the property selection phase. Figure 1 shows the graphical user interface used in the experiment.

The object selection phase occurs in the upper part of the screen. A rectangle with black outline, which occupies 75% of both the screen width and height, displays a 3 by 3 grid of objects. Each object is defined by a combination of four properties: 1/ its position on the grid, 2/ its geometrical shape, 3/ its color, and 4/ a unique identifier represented by a tuple letter-number. At any time, a red ellipse informs which object of the grid is currently selected. If no object is selected the red ellipse is not displayed. For the object selection phase, users are asked to select the object from the grid corresponding to a specific identifier.

The property selection phase occurs in the lower part of the screen. Once an object of the grid is selected, a two-level linear hierarchical menu opens on the lower part of the screen. The position of the menu is independent of the object selected in the first phase. The menu, which occupies 100% of the screen width and 25% of its height, consists of two rows of nine items. Each item is a dark green rectangle with a white text label. The first level items correspond to object properties (e.g., color, size). The second level items, which are updated according to the currently selected first level item, correspond to possible values of a given object property (e.g., blue, red or orange for the object color property). At any time a light green background informs which item of both levels is currently selected. When the menu opens, the central item of both levels is selected by default. In the property selection phase, users validate (i.e., confirm a selection) a particular value (i.e., a second level item) of a given property (i.e., a first level item). Once a value is confirmed, the menu is closed and the object is modified accordingly. If the user starts the object selection phase again, the menu is closed without any value being validated.

The instructions are given textually to the users. Each instruction is composed of an object identifier, an object property to modify and a new value for that property (e.g., `R8: Object Size 1` which instructs users to change the size of the R8 object to the value 1). Instructions are displayed in between the grid and the menu to minimize the visual disruption when users switch their focus between the interactive parts and the instructions.

3.2 Interaction Techniques

To perform the low-priority task, we implemented two interaction techniques: 1/ Baseline, the baseline interaction technique directly inspired from current cockpit interaction[1] and 2/ M[eye]cro, a new interaction technique based on eye-gaze and thumb-to-finger microgestures. Both techniques are performed with the left hand, as the right hand always needs to be ready to grab the side-stick governing the aircraft orientation. In our setup, the side-stick is represented using a handle (i.e., joystick) comfortably placed on the right-hand side of the user. Figure 2 presents the two interaction techniques.

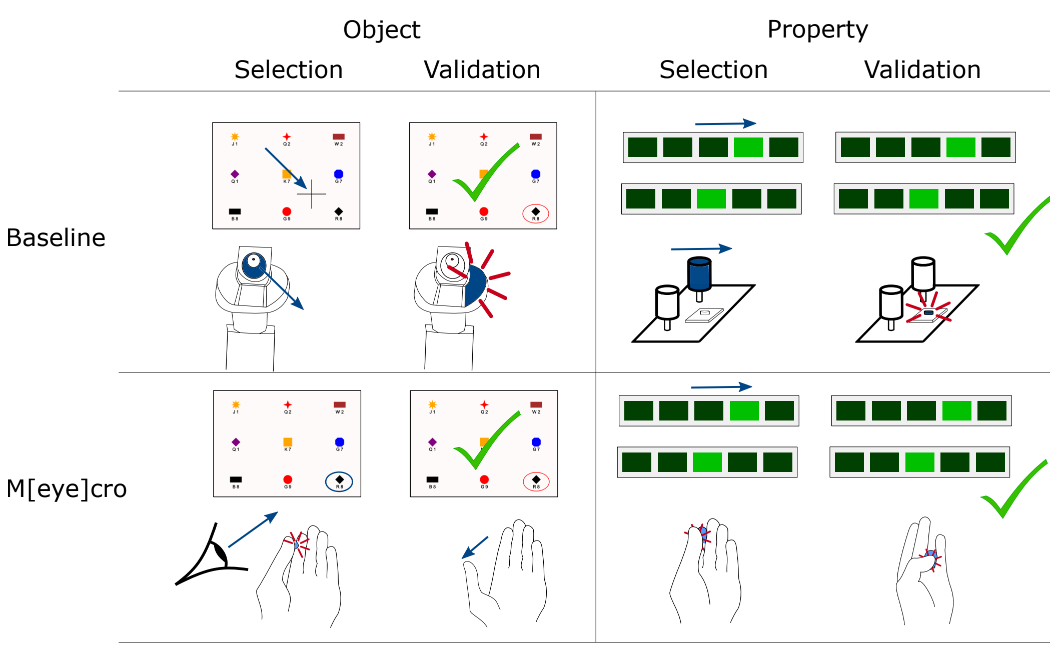

Fig. 2. The two compared techniques: interaction during the two phases of object and property selection. Baseline is the interaction technique directly inspired from current cockpit interactions. It uses a handle composed of a *Hat Switch* to move a cursor in the object selection phase and button to validate the object. In the property selection phase, it uses two *Incremental Rotary Encoders* to move a cursor sideways in each menu level and a physical button to validate the selected property. M[eye]cro is a new interaction technique using eye-gaze and a thumb-to-finger tap to select an object, phalanx taps to navigate both menu levels and a nail tap to validate the property.

[1]We implemented the Baseline interaction technique based on descriptions made by aviation professionals.

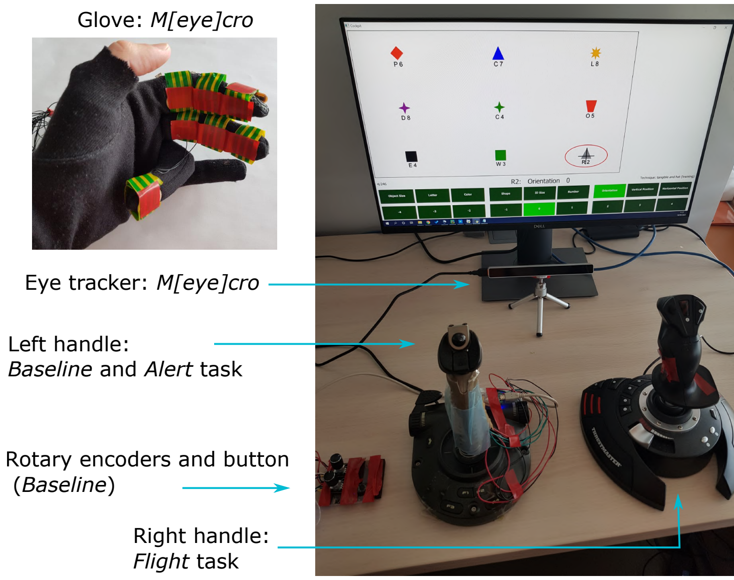

Fig. 3. Glove used by the M[EYE]CRO technique and setup of the experiment.

3.2.1 BASELINE.

BASELINE is the baseline interaction technique we tested, directly inspired from current cockpit interactions. We chose to base this technique on physical controllers as they are still commonly used in cockpits, especially for military aircrafts [44]. Moreover their performance tends to be better than touchscreens in a cockpit context, i.e., when the pilot has to look elsewhere or in the case of degraded conditions such as turbulence [12, 13, 22, 48]. BASELINE uses a *Hat Switch*[2] and two *Incremental Rotary Encoders*[3]. A *Hat Switch* is an isometric pointing device resembling a *Trackpoint*. An *Incremental Rotary Encoder* is a physical knob that can turn endlessly in a discrete manner in both rotational directions. In our setting, shown in Figure 3, a handle (i.e., joystick) with a hat switch atop is placed to the left of the user. Two incremental rotary encoders are placed in a column on a horizontal surface 10cm on the left of the handle and firmly fixed to the desk plane.

In the object selection phase users move a cross-hair using the hat switch. To increase the visibility of the cross-hair, we have set its size to twice the size of an object. When the hat switch is pushed in a particular direction, the cross-hair is moved at a constant speed of 22cm/s. When the hat switch is in its resting position, the cross-hair is automatically snapped to the middle of the closest object. At the beginning of a task, the cross-hair is placed on the central object. To confirm the selection of an object, users press a physical button located atop the handle on the right of the hat switch. Once the confirmation button has been pressed, the menu opens and the next phase begins.

In the property selection phase users move a cursor in each of the menu levels using a dedicated rotary encoder. The upper (respectively lower) rotary encoder moves the cursor of the first (resp. second) level menu. A clockwise (resp. counter-clockwise) rotation of an encoder moves the corresponding cursor rightwards (resp. leftwards). Cursors are blocked on both ends of the menus. To validate a second level item, users press a physical button placed on the desk next to the two encoders.

[2]https://en.wikipedia.org/wiki/Joystick#Hat_switch
[3]https://en.wikipedia.org/wiki/Rotary_encoder#Incremental_encoder

3.2.2 M[EYE]CRO.

M[EYE]CRO is a new interaction technique based on eye-gaze and thumb-to-finger microgestures. M[EYE]CRO is designed to rely on proprioception of users' hand. M[EYE]CRO uses an Eye Tribe© eye tracker and a left-handed glove prototype capable of detecting thumb-to-finger microgestures (i.e., taps of the thumb on the other fingers' phalanxes and nails). Similarly to BASELINE, a handle is placed left of the user whose purpose will be explained in the high-priority task section.

Our glove prototype, shown in Figure 3, designed to capture simple thumb-to-finger microgestures, is a common glove made of a thin cloth. The tip of the thumb is cut out allowing smoother gestures. We placed 18.5mm wide circular Force-Sensitive Resistors (FSR)[4] to detect thumb taps on different parts of the left hand. Following comfort and preference recommendations of previous studies [18, 21, 36, 46], we placed a total of 8 sensors: three sensors on each phalanx inner side of the index and middle fingers, one sensor on the ring finger nail and one sensor on the lateral side of the upper index finger phalanx[5]. We also used flex sensors[6] placed on the back of the index and middle fingers sensing how bent fingers are. The flex sensors are used to disable the FSRs and avoid false trigger when the hand is grabbing the handle. The sensors values are read by an Arduino Micro[7], and then sent to our application which derives the corresponding interaction inputs (e.g., tap on the ring fingernail).

In the object selection phase users enable the eye tracker by pressing down the FSR sensor on the lateral side of the upper index finger. When enabled, users select a target with their eye-gaze. A black ellipse circles the closest object from the eye-gaze location on the screen. To keep the method simple and avoid external parameters we do not use a disambiguation mechanism as objects are distant enough from each others. To confirm the selection of an object, users simply release their press on the FSR sensor. Once confirmed, the menu opens and the next phase begins. This technique allows users to explicitly control the selection mode during which the eye tracker is activated. It avoids the Midas touch problem and any visual perturbations when the eye tracker is not activated.

In the property selection phase users move a cursor in each of the menu level using taps on the index and middle finger phalanxes. A tap on the lower (respectively upper) phalanx of the index finger moves the first level menu cursor leftwards (resp. rightwards). Similarly, taps on the middle finger phalanxes move the second level menu cursor. Long taps (i.e., taps lasting over 300ms) move the cursor continuously at a pace of 6.5 items/s (empirically determined through informal pilot testing) in the appropriate direction. Central phalanxes behave as the lower ones, as informal pilot testing highlighted that tapping on the lower phalanxes was more difficult than tapping on the upper ones. Cursors are blocked on both ends of the menus. To validate a second level item, users press the ring fingernail FSR sensor.

3.3 High-Priority Tasks

The context of an aircraft cockpit involves task switching and multitasking. When performing the task described above, pilots still need to be aware of their environment. They may be interrupted by an alarm that they need to acknowledge or treat, and depending on the flight phase they may need to keep an explicit control of the aircraft orientation. In order to study task switching and multitasking, we introduced different high-priority tasks: 1/ NOSECONDARY which acts as a control (i.e., users only focus on the low-priority task), 2/ ALERT which interrupts the low-priority task requiring users to quickly take an action when being prompted, and 3/ FLIGHT which forces users to continuously monitor and correct their aircraft orientation while performing the low-priority

[4] https://www.adafruit.com/product/166
[5] https://en.wikipedia.org/wiki/Anatomical_terms_of_location#Medial_and_lateral
[6] https://www.adafruit.com/product/1070
[7] https://store.arduino.cc/arduino-micro

task. These two high-priority tasks are designed to force users to treat them. If not treated quickly or precisely enough, the low-priority task is labelled as failed and stopped. Figure 1 shows the graphical user interfaces of the two high-priority tasks.

3.3.1 Alert.

For the Alert, we use a visual stimuli to interrupt users: a green square placed on the right side of the screen suddenly turns red prompting the users to treat it within one second. To treat it, users need to squeeze the trigger-button of the handle with their left hand. If squeezed too late, the low-priority task is stopped. The moment at which the alert is prompted is controlled in the experimental design.

3.3.2 Flight.

For the Flight, we use an aircraft piloting task in the manner of MatB-II [11]. On the right side of the screen, an abstract flight instrument represented by a graduated square with a central area is displayed. A reticle (represented by a circle with a central dot) controlled by the side-stick (i.e., the right-hand side handle) is shown inside this instrument. Users' goal is to keep the reticle within the central area. The reticle is speed-controlled. If the reticle is outside the central area, an internal timer of three seconds is drained down. If the timer runs out, the low-priority task is stopped. When the reticle is inside the central area, the timer is gradually filled up, never exceeding three seconds. To complicate the task, we add random motions to the reticle composed of a random noisy movement (instant displacement of small magnitude, less than 3cm)) and a random continuous drift (i.e., continuous smooth movements of large magnitude, 3cm/s during 0.1 to 2 seconds). Drifts appear every 1 to 5 seconds.

4 EXPERIMENT

We have two goals with this experiment: 1/ comparing the raw performances between M[eye]cro and Baseline (i.e., only focusing on the low-priority task), and 2/ comparing the performances between M[eye]cro and Baseline in the case of multitasking (i.e., with high-priority tasks).

We formulate two hypotheses:

- H_1 Without any high-priority tasks, there is no difference of performance between M[eye]cro and Baseline.
- H_2 M[eye]cro yields better performance than Baseline with high-priority tasks.

4.1 Apparatus

Participants were comfortably sited in front of a desk. A 27-inch display was placed on the desk 80cm in front of the participants. The experiment software ran on a 2018 Dell Latitude 5490 Windows 10 computer. The software interface was displayed in full-screen mode. Both handles (i.e., joysticks), presented in the previous section, were situated on the desk. Their positions were adjusted to better fit each participant morphology and fixed thanks to suction pads.

4.2 Study Design and Procedure

Participants were welcomed by the experimenter who gathered demographic data and presented the experiment. The experimenter then showed and explained the high-priority tasks. Participants trained on the high-priority tasks alone, treating 10 alerts successfully and performing 30 seconds of successful reticle control. Participants then completed the experiment one technique at a time. Before each technique, the experimenter presented the interactions. When using the Baseline technique, they did not wear the glove. For both Baseline and M[eye]cro, they first tried the technique on a menu alone to get used to it. At the end of each condition (i.e., tuple technique/high-priority task), participants filled a raw Nasa TLX questionnaire. At the end of the experiment, we performed a semi-structured interview and we asked participants to rank techniques per condition, to fill a questionnaire on their preferences. They were also free to make any comment on the experiment to the experimenter.

We used a within-subject experimental design in which we tested two factors: TECHNIQUES and TASKTYPES. We had two TECHNIQUES: 1/ BASELINE and 2/ M[EYE]CRO. We had three TASKTYPES: 1/ NOSECONDARY, 2/ ALERT and 3/ FLIGHT. For each combination of TECHNIQUE and TASKTYPE, participants completed five BLOCKS in a row: one training BLOCK, and four experimental BLOCKS. Each BLOCK was composed of nine TRIALS, except for the training BLOCK which had five. A TRIAL is an instance of one low-priority task. In addition to possible high-priority tasks failures, a TRIAL fails when a wrong selection is confirmed by validating a second level item. For this there is no possible correction. In total, we logged 2 TECHNIQUES × 3 TASKTYPES × 4 BLOCKS × 9 TRIALS = 216 TRIALS per participant.

Participants completed all the TRIALS using a technique before moving on to the next one. The order of TECHNIQUES was counter-balanced across participants. NOSECONDARY was always first. Half of the participants had ALERT always second and FLIGHT always third, the other half did FLIGHT second and ALERT third.

In each BLOCK, the nine tasks covered all nine possible object positions, all nine possible first level menu items, and all nine possible second level menu items. Combinations were chosen pseudo-randomly and were all different (i.e., no combination was repeated twice in the entire experiment). The geometrical shape, the color and the unique identifier of each object were chosen randomly for each TRIAL. The first level items were category names in a different random order for each TRIAL to avoid participants learning the ordering of the menu items. The second level items were ordered numeric values (related to the category). We ensured that the first level item that needed to be selected was always a category leading to numeric values (i.e., ID Size, Number, Vertical Position, Horizontal Position Orientation, and Object Size), to test both menu types (i.e., un-ordered for the first level and ordered for the second).

For the ALERT condition, alerts could be triggered during the object selection phase, during the property selection phase or between these two phases. Only one alert occurred during a TRIAL. Within a BLOCK, three TRIALS were randomly assigned to each of the alert phases.

At the beginning of each TRIAL, an instruction screen prompted participants to position their hand(s) accordingly (i.e., left hand on the left handle, and during the FLIGHT condition their right hand on the right handle as well). The left handle was equipped with light sensors allowing the system to track whether the hand was on or off the handle. Once participants had the correct hand(s) position, the TRIAL started displaying the interface and the current task instruction. All input and output events were logged and timed from the beginning of the TRIAL. A TRIAL ended when the task was completed and the left hand was back on its handle.

A TRIAL was considered erroneous: if the wrong second level menu item was validated, if an alert was not treated quickly enough in the ALERT condition, or if a tracking timer ran out in the FLIGHT condition. An erroneous TRIAL was repeated at the end of the same BLOCK until it was successfully completed.

4.3 Participants

We recruited 12 participants in the local university campus: 6 females, 6 males, mean age 27.8 (4.5 STD), median age 27.5, all right-handed. All participants had previously used the glove in another experiment. Our goal was to minimize the *wow factor* from first time use as well as maximize participants' familiarity with thumb-to-finger microgestures.

4.4 Results

For all results we follow Dragicevic's advices on statistical communication for HCI [14]. Therefore, we use estimation methods to derive 95% confidence intervals (CIs) rather than traditional null hypothesis statistical testing.

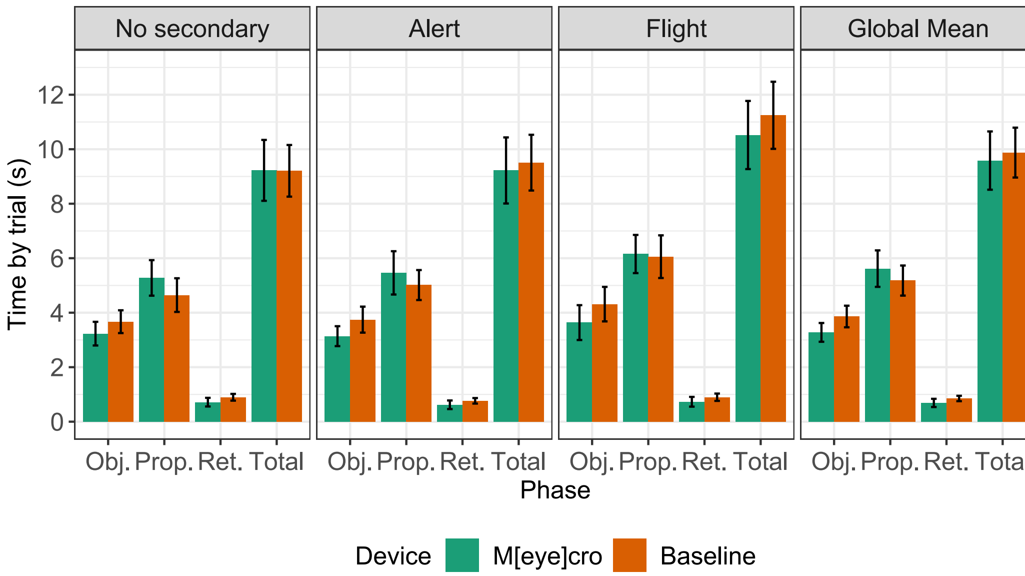

Fig. 4. Mean trial completion time and CIs for each high-priority task condition, each trial phase (i.e., *object* selection, *property* selection, *return* (phase during which participants get their left hand back on the handle), each technique and each high-priority task.

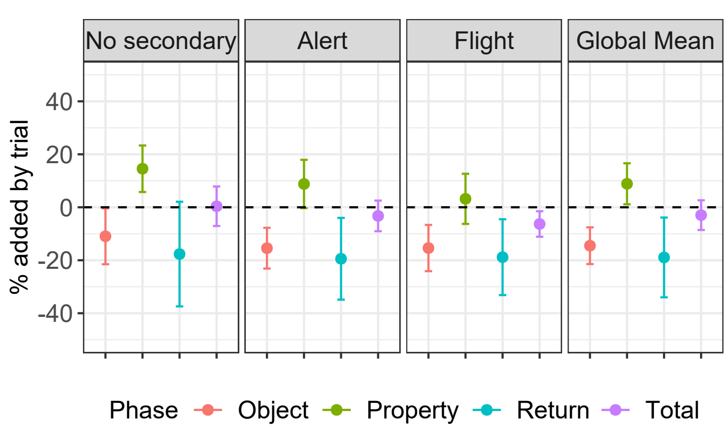

Fig. 5. Participants' mean trial completion time ratio and CIs for each technique, phase and high-priority task. Values correspond to the percentage of added trial completion time for M[EYE]CRO compared to BASELINE (i.e., negative values mean that M[EYE]CRO is faster).

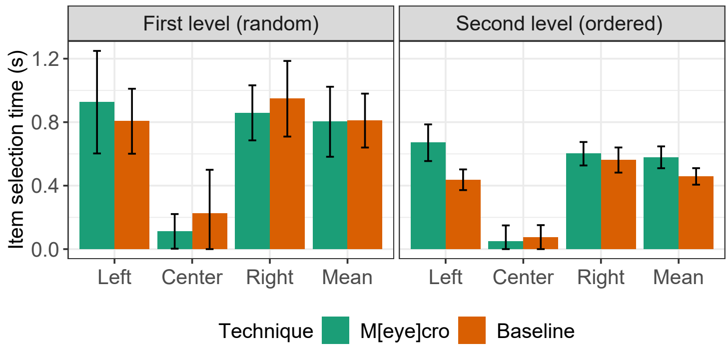

Fig. 6. Mean item selection time and CIs for each technique, each menu level (first-level – random item order, second-level – ordered items) and item position in the menu.

4.4.1 Completion time. Figure 4 shows the mean and CIs of the total trial time for each high-priority task condition, each trial phase (i.e., *object* selection, *property* selection, *return* (phase in which participants get their left hand back on the handle) and each technique. Figure 4 also shows the global mean time for each phase and technique across all high-priority task conditions. For an extended analysis we also present participants' mean time ratio between M[EYE]CRO and BASELINE (Figure 5), where values correspond to the percentage of added trial completion time for M[EYE]CRO compared to BASELINE (i.e., negative values mean that M[EYE]CRO is faster).

In the NOSECONDARY condition the mean total trial time is similar between the two techniques (M[EYE]CRO 9.22s CI [8.11, 10.34], BASELINE 9.21s CI [8.26, 10.16]). For the object selection phase, results suggest that M[EYE]CRO (3.23s CI [2.80, 3.67]) is faster than BASELINE (3.67s CI [3.25, 4.09]) however, BASELINE seems to be faster for the property selection phase (M[EYE]CRO 5.28s CI [4.63, 5.93], BASELINE 4.65s CI [4.03, 5.26]). In the ALERT condition, we observe a similar trend for the phases but the global time ratio suggest that M[EYE]CRO (9.22s CI [8.01, 10.43]) is faster than BASELINE (9.51s CI [8.48, 10.53]). In the FLIGHT condition, time ratios suggest that M[EYE]CRO (10.52s CI [9.27, 11.77]) is faster than BASELINE (11.25s CI [10.01, 12.48]). It could be explained by the smaller difference between both techniques in the property selection phase. Even though, the global mean time across all conditions seems to be slightly in favor of M[EYE]CRO, we cannot conclude on a faster technique given the CIs. However, we can note that the return phase seems to be always faster for M[EYE]CRO but given its magnitude (less than half a second) this difference has little to no impact on the total time.

Figure 6 shows the mean item selection time with both techniques and each menu level (i.e., first-level with items in a random order, and second-level with items ordered). It also breaks down the mean item selection time between the position of the items (i.e., left parts of the menu combined, central position alone or right parts of the menu combined). As expected, the selection times for the second level (M[EYE]CRO 0.58s CI [0.51, 0.65], BASELINE 0.46s CI [0.41, 0.51]) are faster than for the first level (M[EYE]CRO 0.80s CI [0.58, 1.02], BASELINE 0.81s CI [0.64, 0.98]) due to the different natures of item ordering. We saw previously that BASELINE seems globally faster than M[EYE]CRO in this phase, it seems that this difference is due to the second level menu item selection for which BASELINE performed better in the case of an ordered list of items.

4.4.2 Errors. Figure 7 shows the average number of errors for each high-priority tasks and techniques. The figure also shows error types: non treated alerts, failed FLIGHT, wrong selected object,

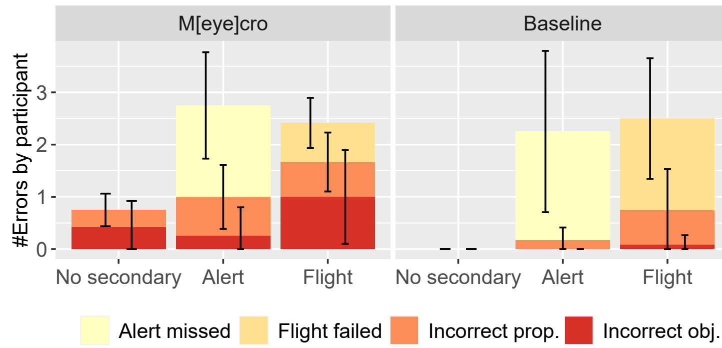

Fig. 7. Average number of errors by participant for each high-priority task and technique. Errors are broken down into their different types (prop. stands for property and obj. for object). CIs are associated to error types.

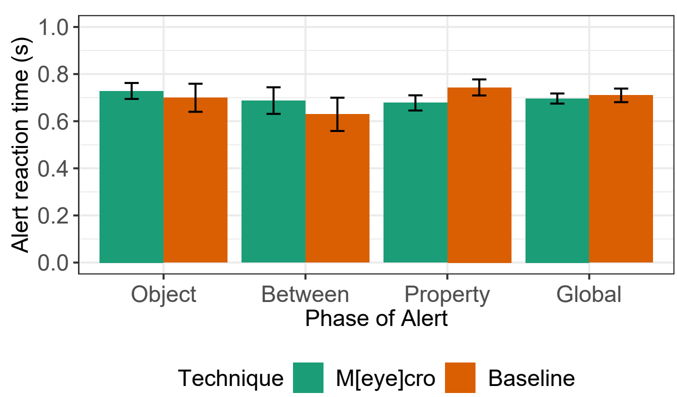

Fig. 8. Mean alert reaction time and CIs for each technique and each alert trigger phase.

wrong selected property. M[EYE]CRO seems to induce more object and property selection errors for all high-priority tasks. After further analysis, most object selection errors are due to participants' eye-gaze moving to another part of the interface while validating, changing the object that users wanted to originally select. For property selection errors, we hypothesize that accidental triggering of the continuous cursor movement in the menu (i.e., dwell tap) could be the cause. However, BASELINE yields more alert and tracking failures than M[EYE]CRO.

4.4.3 High-priority tasks. Figure 8 shows the mean alert reaction time for each technique and each phase in which alerts were triggered. For an extended analysis we also present the participant mean alert reaction time ratio between M[EYE]CRO and BASELINE (Figure 9), where values correspond to the percentage of added reaction time for M[EYE]CRO compared to BASELINE (i.e., negative values mean that M[EYE]CRO is faster). Overall, reaction times are pretty similar for both techniques. As expected in the object selection phase, both techniques allow us to perform interaction very close to the alert button, meaning that the motor movement time needed to press the handle button is very negligible compared to the mental process. The same is true in-between the phases, as the left hand is not yet too far from the handle in the BASELINE condition. However, a small advantage (around

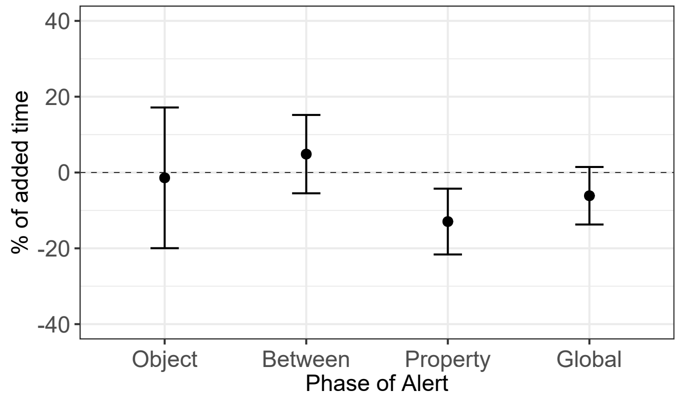

Fig. 9. Mean alert reaction time ratio and CIs between M[eye]cro and Baseline. Values correspond to the percentage of added reaction time for M[eye]cro compared to Baseline (i.e., negative values mean that M[eye]cro is faster).

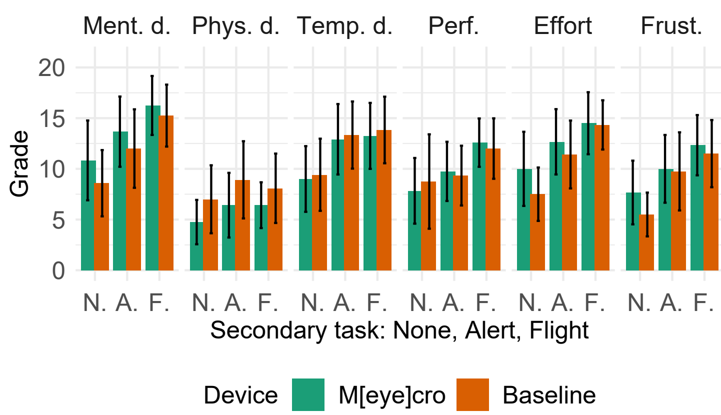

Fig. 10. Results of the raw NASA TLX (Mean and CI) per high-priority task: mental demand, physical demand, temporal demand, performance, effort and frustration. The higher the value, the most negative the answer, e.g., a higher performance grade means less performance.

10% faster) for M[eye]cro can be observed in the property selection phase as, for Baseline, the left hand is deported onto the rotary encoders whereas the hand always stays close to the handle for M[eye]cro.

For the Flight condition both techniques led to similar performances in terms of percentage of time spent in the target square (M[eye]cro 81.4% CI[79.3, 83.4], Baseline 81.5% CI[79.2, 83.9]).

4.4.4 Qualitative results. Figure 10 shows the mean Raw Nasa TLX score by high-priority task and technique. Figure 11 highlights the difference in the TLX scores per participant between Baseline and M[eye]cro, where a negative (resp. positive) value shows results in favor of M[eye]cro (resp. Baseline). The Raw Nasa TLX questions were 21 Likert scales. There are no major differences between both techniques. We can note that M[eye]cro seems to yield a slightly higher mental

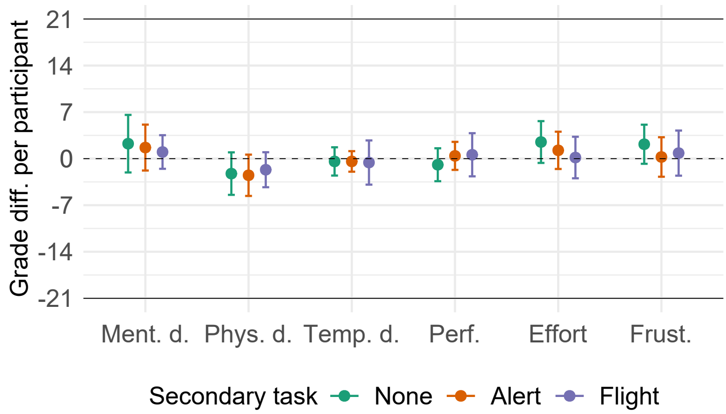

Fig. 11. Mean and CI of the difference per participant for each TLX scale between the grades of M[EYE]CRO and BASELINE. A negative value means that M[EYE]CRO is preferred.

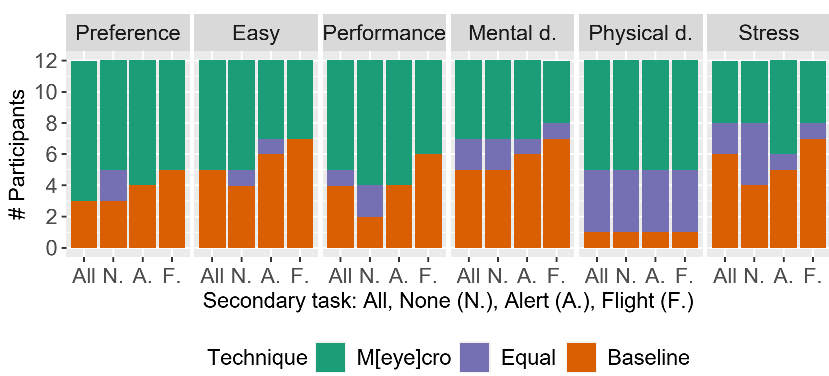

Fig. 12. Distribution of answers for participants' preferences, perceived ease of use, perceived performance, mental demand, physical demand and perceived stress for each technique and high-priority task.

demand, effort and frustration in the NOSECONDARY condition. This is also true for the other conditions but with a more negligible effect.

Figure 12 shows the distribution of answers for participants' preferences, perceived ease of use, perceived performance, mental demand, physical demand and perceived stress for each high-priority task. Overall, M[EYE]CRO was preferred regardless of the high-priority task condition. It also yielded less perceived physical demand (which can also be seen in the Raw Nasa TLX answers), and a better perceived performance (even though it is not confirmed by the Raw Nasa TLX answers).

5 DISCUSSION

We underline that our goal is not to assess whether the glove will work in on-flight conditions but rather if there are potential benefits in using the gestures it senses. If compelling results are found, follow up projects on hardware design can then be initiated.

Overall, when used with no high-priority task, M[EYE]CRO seems to yield similar performances than those of BASELINE. This result is a valuable starting point to explore an alternative option to

Baseline. Moreover for multitasking, i.e., when performing an additional task in parallel, results suggest that time performances with M[eye]cro seem better than with Baseline while exhibiting more errors (around 1 error for M[eye]cro vs. 0 for Baseline per 36 trials). In the Alert condition the two techniques perform equally, showing that M[eye]cro can be interrupted by an external stimulus, however, the reaction time to the stimulus was faster for M[eye]cro. In the Flight condition M[eye]cro seems to yield better performances in terms of low-priority task completion time and also less tracking errors. For both conditions, we hypothesize that M[eye]cro drains less users' visual attention. Moreover, Baseline induces more alert and tracking failures. For all these reasons, M[eye]cro seems to be a promising new technique to perform tasks in parallel.

Unsurprisingly, time benefits of M[eye]cro mostly come from the object selection phase, since users will always visually search the target first regardless of the technique. This gives a clear advantage to selections based on eye-gazing. Time performances also demonstrate that the combination of eye-gaze and thumb-to-finger is a feasible and reliable solution. Time benefits also come from the return phase, since the hand performing microgestures can stay close to the handle. We hypothesize that time differences in the return phase might even be more pronounced depending on the knobs' placement. The distance to cover in order to grab the physical knobs will vary according to their positions relative to the handle, whereas microgestures are always *at hand* (assuming that the sensing technology is always available e.g., camera always on and not occluded, glove always on and worn).

In the property selection phase, Baseline is faster than M[eye]cro, even though in the Flight condition both techniques yield similar time performances. The difference comes from the second level menu (i.e., ordered list of items). We hypothesize that to input a known sequence of discrete actions, physical controllers with haptic feedback (e.g., clicks of the rotary encoders) are better suited than only relying on users' proprioception. M[eye]cro also induces more property selection errors than Baseline. Those poor performances could be due to our prototype (e.g., accidental trigger of continuous cursor movement). Moreover, caution while operating the prototype could also account for those time differences between M[eye]cro and Baseline. Reducing errors during the property selection phase is critical. To do so we need to investigate if a more robust version of the glove would solve this problem.

Moreover, a drawback of using eye-gaze to select objects is the potential interference with other tasks requiring visual attention, as we observed in the error analysis. Solutions exist to account for this problem, such as analyzing which part of the interface is looked at, as in GazeForm [31], or account for the type of gaze movements like detecting saccades [16]. Given the relatively higher number of errors in the object selection phase and participants' feedback (e.g., several participants stated that they were still learning while running through the experiment), we hypothesize that the performances of M[eye]cro could be improved with practice.

Finally, another limitation is the grid design during the object selection phase of the low-priority task. The grid was large and objects were sufficiently spaced out. These are perfect conditions for eye-gaze pointing as eye-tracking errors are smaller than the distance between the objects. To account for small or packed objects, a more accurate eye-tracker or pointing technique with a disambiguation phase (largely studied in the literature [27]) would be necessary. It is worth noticing that spaced out objects also give an edge to the snapping mechanism of Baseline.

M[eye]cro allows users to select on-screen objects and menu items. Such basic interaction tasks are common to several contexts but the generalization of the obtained results require more studies. While the measured performance is valid in the context of a cockpit interaction standard, other contexts have different baseline techniques with their own benefits and drawbacks. However, our results quantitatively established that thumb-to-finger microgestures are worth exploring.

6 CONCLUSION

Thumb-to-finger microgestures are gestures performed with the thumb of a hand onto the fingers of the same hand. In this paper we study the use of thumb-to-finger gestures in an integrated interaction technique for object and command selection and we test the state-of-the-art hypothesis stating that thumb-to-finger microgestures could be beneficial for multitasking. To do so we designed M[eye]cro, a new interaction technique to select on-screen objects and navigate menus through the synergistic use of eye-gaze and thumb-to-finger microgestures. In the context of cockpit interaction, we compared M[eye]cro to an on-board aircraft interaction technique based on physical controllers in a controlled experiment. This experiment is the first to evaluate the potential of thumb-to-finger microgestures for multitasking.

The experiment considers the low-priority task of object and command selection but also task switching with a high-priority alert monitoring task as well as multitasking with a high-priority continuous retical control task. Results show that without any high-priority tasks the performances are similar for both techniques, with M[EYE]CRO performing better in the object selection phase, but poorer in the command selection. However, for task switching and multitasking M[EYE]CRO tends to yield better time performance for parallel tasks (flight), and in 1 out of 3 interrupt cases (alert-property, Figure 9). High-priority tasks are also better performed (smaller reaction time in the alert condition, and less tracking error in the continuous control condition). Finally, M[EYE]CRO induces less fatigue and is mostly preferred by the participants. Yet M[EYE]CRO also leads to more selection errors. To reduce errors we identified two aspects that require further study: the glove prototype robustness and user practice. Nevertheless, these results are promising and reveal valuable insights into the usage of thumb-to-finger microgestures for task switching and multitasking.

ACKNOWLEDGMENTS

The authors thank the DGA (French MoD) and Thales who provide funding support for this study and the "Man-Machine Teaming" scientific program in which this research project is undertaken.

REFERENCES

[1] Sunggeun Ahn and Geehyuk Lee. 2019. Gaze-Assisted Typing for Smart Glasses. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 857–869. https://doi.org/10.1145/3332165.3347883

[2] Alexandre Alapetite, Rune Fogh, David Zammit-Mangion, Christian Zammit, Ian Agius, Marco Fabbri, Marco Pregnolato, and Loïc Becouarn. 2012. Direct tactile manipulation of the flight plan in a modern aircraft cockpit. *Proceedings of the International Conference on Human-Computer Interaction in Aerospace, HCI Aero'2012*.

[3] Renaud Blanch and Michael Ortega. 2009. Rake Cursor: Improving Pointing Performance with Concurrent Input Channels. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI 2009)*. Boston, MA, United States, 1415–1418. https://doi.org/10.1145/1518701.1518914

[4] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[5] Doug A. Bowman and Chadwick A. Wingrave. 2001. Design and Evaluation of Menu Systems for Immersive Virtual Environments. In *Proceedings of the Virtual Reality 2001 Conference (VR'01) (VR '01)*. IEEE Computer Society, USA, 149.

[6] Maurizio Caon, Rico Süsse, Benoit Grelier, Omar Abou Khaled, and Elena Mugellini. 2019. Gesturing on the Handlebar: A User-Elicitation Study for On-Bike Gestural Interaction. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, Sebastiano Bagnara, Riccardo Tartaglia, Sara Albolino, Thomas Alexander, and Yushi Fujita (Eds.). Springer International Publishing, Cham, 429–439.

[7] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 3403–3414. https://doi.org/10.1145/2858036.2858589

[8] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. ACM, New York, NY, USA, 255–260. https://doi.org/10.1145/2501988.2502016

[9] Andy Cockburn, Carl Gutwin, Philippe Palanque, Yannick Deleris, Catherine Trask, Ashley Coveney, Marcus Yung, and Karon MacLean. 2017. Turbulent Touch: Touchscreen Input for Cockpit Flight Displays. 6742–6753. https://doi.org/10.1145/3025453.3025584

[10] A. Cockburn, D. Masson, C. Gutwin, P. Palanque, A. Goguey, M. Yung, C. Gris, and C. Trask. 2019. Design and evaluation of braced touch for touchscreen input stabilisation. *International Journal of Human-Computer Studies* 122 (2019), 21 – 37. https://doi.org/10.1016/j.ijhcs.2018.08.005

[11] James R. Comstock, Ruth J. Arnegard, and Langley Research Center. 1992. *The multi-attribute task battery for human operator workload and strategic behavior research.* National Aeronautics and Space Administration, Langley Research Center.

[12] Louise V. Coutts, Katherine L. Plant, Mark Smith, Luke Bolton, Katie J. Parnell, James Arnold, and Neville A. Stanton. 2019. Future technology on the flight deck: assessing the use of touchscreens in vibration environments. *Ergonomics* 62, 2 (2019), 286–304. https://doi.org/10.1080/00140139.2018.1552013 PMID: 30470162.

[13] Sonia Dodd, Jeff Lancaster, Andrew Miranda, Steve Grothe, Bob DeMers, and Bill Rogers. 2014. Touch Screens on the Flight Deck: The Impact of Touch Target Size, Spacing, Touch Technology and Turbulence on Pilot Performance. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 58, 1 (2014), 6–10. https://doi.org/10.1177/1541931214581002

[14] Pierre Dragicevic. 2016. Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI.* Springer, 291 – 330. https://doi.org/10.1007/978-3-319-26633-6_13

[15] Ribel Fares, Shaomin Fang, and Oleg Komogortsev. 2013. Can We Beat the Mouse with MAGIC?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1387–1390. https://doi.org/10.1145/2470654.2466183

[16] Mackenzie G. Glaholt. 2014. *Eye Tracking in the Cockpit: a Review of the Relationships between Eye Movements and the Aviators Cognitive State.* Technical Report.

[17] John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Itoh, and Satoru Mashino. 2003. *Command Without a Click : Dwell Time Typing by Mouse and Gaze Selections.* IOS Press, 121–128. http://www.hci-international.org/index.php?module=conference&CF_op=view&CF_id=3 10th International Conference on Human-Computer Interaction, HCI International 2003 ; Conference date: 22-06-2003 Through 27-06-2003.

[18] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 1526–1537. https://doi.org/10.1145/2858036.2858483

[19] R. Häuslschmid, B. Menrad, and A. Butz. 2015. Freehand vs. micro gestures in the car: Driving performance and user experience. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 159–160. https://doi.org/10.1109/3DUI.2015.7131749

[20] S. Jalaliniya, D. Mardanbeigi, T. Pederson, and D. W. Hansen. 2014. Head and Eye Movement as Pointing Modalities for Eyewear Computers. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops*. 50–53. https://doi.org/10.1109/BSN.Workshops.2014.14

[21] HaiYan Jiang, Dongdong Weng, Zhenliang Zhang, and Feng Chen. 2019. HiFinger: One-Handed Text Entry Technique for Virtual Environments Based on Touches between Fingers. *Sensors* 19, 14. https://doi.org/10.3390/s19143063

[22] S. Kaminani. 2011. Human computer interaction issues with touch screen interfaces in the flight deck. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. 6B4–1–6B4–7. https://doi.org/10.1109/DASC.2011.6096098

[23] Hsin-Liu (Cindy) Kao, Artem Dementyev, Joseph A. Paradiso, and Chris Schmandt. 2015. NailO: Fingernails As an Input Surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 3015–3018. https://doi.org/10.1145/2702123.2702572

[24] Masatomo Kobayashi and Takeo Igarashi. 2008. Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 949–958. https://doi.org/10.1145/1357054.1357201

[25] Manu Kumar, Andreas Paepcke, and Terry Winograd. 2007. EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 421–430. https://doi.org/10.1145/1240624.1240692

[26] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-Free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1952–1956. https://doi.org/10.1145/2858036.2858335

[27] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655

[28] 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 94 (Sept. 2019), 27 pages. https://doi.org/10.1145/3351252

[29] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: A wearable device for one-handed microinteractions. In *Proc. MobileHCI Workshop on Ensembles and On-body Devices*.

[30] Francisco Lopez Luro and Veronica Sundstedt. 2019. A Comparative Study of Eye Tracking and Hand Controller for Aiming Tasks in Virtual Reality. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 68, 9 pages. https://doi.org/10.1145/3317956.3318153

[31] Sylvain Pauchet, Catherine Letondal, Jean-Luc Vinot, Mickaël Causse, Mathieu Cousy, Valentin Becquet, and Guillaume Crouzet. 2018. GazeForm: Dynamic Gaze-Adaptive Touch Surface for Eyes-Free Interaction in Airliner Cockpits. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 1193–1205. https://doi.org/10.1145/3196709.3196712

[32] Sylvain Pauchet, Jean-Luc Vinot, Catherine Letondal, Alexandre Lemort, Claire Lavenir, Timothée Lecomte, Stéphanie Rey, Valentin Becquet, and Guillaume Crouzet. 2019. Multi-Plié: A Linear Foldable and Flattenable Interactive Display to Support Efficiency, Safety and Collaboration. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300384

[33] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2015. Gaze+touch vs. Touch: What's the Trade-off When Using Gaze to Extend Touch to Remote Displays?. In *15th Human-Computer Interaction (INTERACT) (Human-Computer Interaction – INTERACT 2015, Vol. LNCS-9297)*. Bamberg, Germany, 349–367. https://doi.org/10.1007/978-3-319-22668-2_27

[34] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + Pinch Interaction in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/3131277.3132180

[35] Gowdham Prabhakar, Aparna Ramakrishnan, Modiksha Madan, L. R.D. Murthy, Vinay Krishna Sharma, Sachin Deshmukh, and Pradipta Biswas. 2020. Interactive gaze and finger controlled HUD for cars. *Journal on Multimodal User Interfaces* (2020). https://doi.org/10.1007/s12193-019-00316-9

[36] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: An Eyes-free VR/AR Symbolic Input Device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (Edinburgh, Scotland) *(VRST '14)*. ACM, New York, NY, USA, 9–18. https://doi.org/10.1145/2671015.2671029

[37] Yuan Yuan Qian and Robert J. Teather. 2017. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 91–98. https://doi.org/10.1145/3131277.3132182

[38] Wendy A. Rogers, Arthur D. Fisk, Anne Collins McLaughlin, and Richard Pak. 2005. Touch a Screen or Turn a Knob: Choosing the Best Device for the Job. *Human Factors* 47, 2 (2005), 271–288. https://doi.org/10.1518/0018720054679452 PMID: 16170938.

[39] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping Microgestures: Eliciting Single-Hand Microgestures for Handheld Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 402, 13 pages. https://doi.org/10.1145/3290605.3300632

[40] Linda E. Sibert and Robert J. K. Jacob. 2000. Evaluation of Eye Gaze Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 281–288. https://doi.org/10.1145/332040.332445

[41] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (Tokyo, Japan) *(ISS '18)*. ACM, New York, NY, USA, 177–187. https://doi.org/10.1145/3279778.3279799

[42] Neville Stanton, Catherine Harvey, Katherine Plant, and Luke Bolton. 2013. To twist, roll, stroke or poke? A study of input devices for menu navigation in the cockpit. *Ergonomics* 56 (02 2013). https://doi.org/10.1080/00140139.2012.751458

[43] Yanke Tan, Sang Ho Yoon, and Karthik Ramani. 2017. BikeGesture: User Elicitation and Performance of Micro Hand Gesture as Input for Cycling. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA,

2147–2154. https://doi.org/10.1145/3027063.3053075

[44] Peter Thomas. 2018. Performance, Characteristics, and Error Rates of Cursor Control Devices for Aircraft Cockpit Interaction. *International Journal of Human-Computer Studies* 109 (1 Jan. 2018), 41–53. https://doi.org/10.1016/j.ijhcs.2017.08.003 This document is the Accepted Manuscript version of the following article: Peter R. Thomas, 'Performance, Characteristics, and Error Rates of Cursor Control Devices for Aircraft Cockpit Interaction', International Journal of Human-Computer Studies, Vol. 109: 41-53, available online 31 August 2017. Under embargo. Embargo end date: 31 August 2018. Published by Elsevier. © 2017 Elsevier Ltd. All rights reserved.

[45] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016. TouchRing: Subtle and Always-Available Input Using a Multi-Touch Ring. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 891–898. https://doi.org/10.1145/2957265.2961860

[46] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. 2016. ThumbRing: Private Interactions Using One-handed Thumb Motion Input on Finger Segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) *(MobileHCI '16)*. ACM, New York, NY, USA, 791–798. https://doi.org/10.1145/2957265.2961859

[47] Simon Voelker, Andrii Matviienko, Johannes Schöning, and Jan Borchers. 2015. Combining Direct and Indirect Touch Input for Interactive Workspaces Using Gaze Input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (Los Angeles, California, USA) *(SUI '15)*. Association for Computing Machinery, New York, NY, USA, 79–88. https://doi.org/10.1145/2788940.2788949

[48] Simon Voelker, Kjell Ivar Øvergård, Chat Wacharamanotham, and Jan Borchers. 2015. Knobology Revisited: A Comparison of User Performance between Tangible and Virtual Rotary Knobs. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. Association for Computing Machinery, New York, NY, USA, 35–38. https://doi.org/10.1145/2817721.2817725

[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 113 (Sept. 2017), 21 pages. https://doi.org/10.1145/3130978

[50] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 2011. A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction - Volume Part I* (Lisbon, Portugal) *(INTERACT'11)*. Springer-Verlag, Berlin, Heidelberg, 559–575. https://doi.org/10.1007/978-3-642-23774-4_45

[51] Katrin Wolf, Robert Schleicher, Sven Kratz, and Michael Rohs. 2013. Tickle: A Surface-Independent Interaction Technique for Grasp Interfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (Barcelona, Spain) *(TEI '13)*. Association for Computing Machinery, New York, NY, USA, 185–192. https://doi.org/10.1145/2460625.2460654

[52] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9: Leveraging Thumb-to-Finger Interaction for Same-Side-Hand Text Entry on Smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 178, 10 pages. https://doi.org/10.1145/3173574.3173752

[53] Yiqi Xiao and Renke He. 2019. The Handlebar as an Input Field: Evaluating Finger Gestures Designed for Bicycle Riders. In *Advances in Intelligent Systems and Computing, vol 786*, Neville Stanton (Ed.). Springer International Publishing, Cham, 648–659. https://doi.org/10.1007/978-3-319-93885-1_59

[54] Yiqi Xiao and Renke He. 2019. The intuitive grasp interface: design and evaluation of micro-gestures on the steering wheel for driving scenario. *Universal Access in the Information Society* (04 2019), 1–18. https://doi.org/10.1007/s10209-019-00647-0

[55] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 883–899. https://doi.org/10.1145/3332165.3347865

[56] Sang Ho Yoon, Ke Huo, Vinh P. Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-Worn Textile Input Device with Multimodal Sensing in Mobile Interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (Stanford, California, USA) *(TEI '15)*. Association for Computing Machinery, New York, NY, USA, 269–272. https://doi.org/10.1145/2677199.2680560

[57] Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2014. Plex: Finger-worn Textile Sensor for Mobile Interaction During Activities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Seattle, Washington) *(UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 191–194. https://doi.org/10.1145/2638728.2638746

[58] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 246–253. https://doi.org/10.1145/302979.303053

[59] Xuan Zhang and I. Scott MacKenzie. 2007. Evaluating Eye Tracking with ISO 9241 - Part 9. In *Proceedings of the 12th International Conference on Human-Computer Interaction: Intelligent Multimodal Interaction Environments* (Beijing, China) *(HCI'07)*. Springer-Verlag, Berlin, Heidelberg, 779–788. https://doi.org/10.1007/978-3-540-73110-8_85

[60] Qiushi Zhou and Eduardo Velloso. 2017. GazeGrip: Improving Mobile Device Accessibility with Gaze & Grip Interaction. In *Proceedings of the 29th Australian Conference on Computer-Human Interaction* (Brisbane, Queensland, Australia) *(OZCHI '17)*. Association for Computing Machinery, New York, NY, USA, 467–471. https://doi.org/10.1145/3152771.3156159

Received February 2021; accepted April 2021