# Interactive Mediation Techniques for Error-Aware Gesture Input Systems

Rawan Alghofaili
University of Texas at Dallas
Richardson, Texas, USA
rawan@utdallas.edu

Naveen Sendhilnathan
Reality Labs Research, Meta
Redmond, Washington, USA

Ting Zhang
Reality Labs Research, Meta
Redmond, Washington, USA

Tovi Grossman
University of Toronto
Toronto, Ontario, Canada

Michael Glueck
Reality Labs Research, Meta
Toronto, Ontario, Canada

Tanya R. Jonker
Reality Labs Research, Meta
Redmond, Washington, USA

Ben Lafreniere
Reality Labs Research, Meta
Toronto, Ontario, Canada

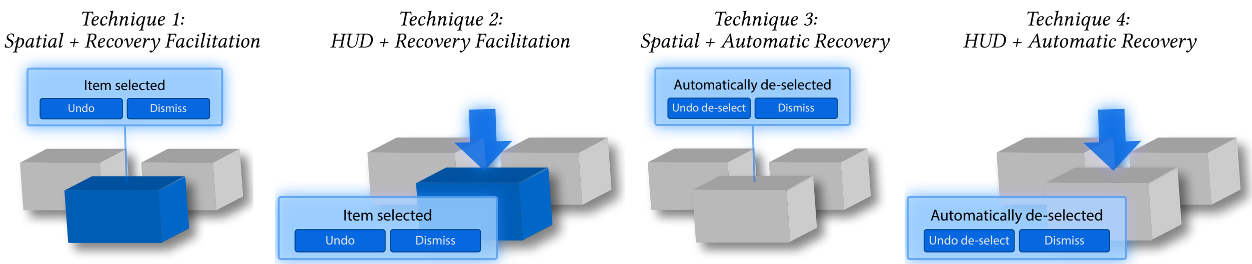



**Figure 1: Four error mediation techniques, organized by notification method (Spatial vs. HUD) and system initiative (Recovery Facilitation vs. Automatic Recovery).**

## ABSTRACT

Input false-positive errors, where a system recognizes an input action that the user did not perform, have been shown to be particularly costly for user experience. Recent work has suggested that eye-gaze behavior immediately following an input event can be used to detect whether the input was intended by a user or was the result of a false-positive error. The ability to detect these errors could enable systems that assist the user with error recovery, but little is currently known about how such *error mediation techniques* might be designed, or the benefits they could provide. This paper presents an initial investigation of the design of error mediation techniques, and an evaluation of their potential benefits. A controlled study demonstrated that error mediation techniques can save time when recovering from errors by helping users to notice and resolve these errors quickly when they occur.

## CCS CONCEPTS

• **Human-centered computing** → **Gestural input**; **Empirical studies in HCI**; **Mixed / augmented reality**.

## KEYWORDS

recognizer error, gaze behavior, eye tracking, adaptive user interfaces

**ACM Reference Format:**
Rawan Alghofaili, Naveen Sendhilnathan, Ting Zhang, Tovi Grossman, Michael Glueck, Tanya R. Jonker, and Ben Lafreniere. 2018. Interactive Mediation Techniques for Error-Aware Gesture Input Systems. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

Augmented reality technologies have the potential to seamlessly integrate interactions with digital content into our everyday activities in the physical world. However, to realize such seamless user experiences, we need input systems that can reliably distinguish intentional input actions (such as performing a mid-air pinch gesture) from other user behaviors (such as picking up a coffee cup). When an input recognizer fails to discriminate between these actions, two types of error can occur: *false positives*, where a system incorrectly


*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

recognizes an input action that the user did not intentionally perform, and *false negatives*, where a system fails to recognize an input action that the user did intentionally perform. While both types of errors have been shown to degrade user experiences, false-positive errors have been shown to be particularly costly [15, 17] due in part to the attentional demands required to notice and fix these errors when they occur [17].

A substantial body of research has explored how to reduce recognition errors by proposing more accurate gesture recognition algorithms [23], by choosing gestures that are unlikely to occur naturally [11, 31], or by adding delimiter gestures [12, 24]. However, many of these approaches trade-off the naturalness of input for a reduction in error rate, and there are limits to how well recognition algorithms can detect gestures that are similar to natural user behaviors.

A complementary approach to reducing the error rate, which is compatible with natural gestures, is to accept that some errors will inevitably occur and design the interface to permit easy error recovery. Supporting error recovery is well established as a fundamental aspect of usability, but a number of recent works have suggested that it may be beneficial for systems to detect and dynamically initiate error recovery interactions with the user, or even automatically roll back input from the user. Schwarz et al. called out the need for such approaches in their work on frameworks handling inputs with uncertainty [26], as did Schmid et al. in their work studying input errors that occur as a result of "last instant" changes to an interface state [25]. More recently, the idea of detecting input errors and assisting with error recovery was proposed in work by Peacock et al., which demonstrated that false positive input errors can be detected from a user's eye-gaze behaviour 50 to 550ms after an input event [21], and in subsequent work by Sendhilnathan et al. which built on this finding to demonstrate that eye-gaze behavior following an input event can be used to distinguish between correct input, input false positive errors, and user errors [28]. Despite being called out repeatedly as an area for investigation, research has not been conducted to understand how exactly a system might best dynamically assist the user upon detecting that a recent input event may have been caused by an error, or to establish the benefits of such dynamic error recovery in terms of error recovery time or other measures.

To answer these questions, we propose a set of designs for interactive *error mediation techniques* that could be powered by an input false-positive detection model. Our investigation focused on two design variables: (1) where users are notified about potential errors (in a heads-up display vs. spatially locked to affected items), and (2) the degree of initiative the system takes to reverse changes to the application state, resulting in the four designs shown in Fig. 1. In a controlled study, we investigate the benefits of these designs against a baseline of no error mediation. Users were asked to complete an object selection task in a simulated AR environment, and input false positive errors were simulated by injecting clicks at random intervals while the user performed the task. Our study results show that error mediation techniques powered by input false-positive detection can save the user time while recovering from errors, and that this advantage is due to their ability to ensure that errors are noticed and resolved rather than being missed. We also found a preference for spatial notification of errors over a heads-up presentation, and mixed support for techniques that automatically correct actions on behalf of the user. Collectively, this work makes the following contributions:

- An investigation of the design of error mediation techniques for assisting a user with recovering from false-positive input errors.
- Quantitative and qualitative results demonstrating the benefits of error mediation and insights into the design of such techniques.
- Design implications for the development of error mediation techniques, including a simple framework for designers to use when thinking through the potential user experiences created by these techniques.

## 2 RELATED WORK

This work extends prior research on gesture recognition errors, interactive mediation techniques, and techniques for highlighting changes in user interfaces.

### 2.1 Gesture Recognition Errors

The goal of gesture recognition algorithms is to support high rates of precision and recall [19, 22, 24]. Lower precision results in more false-positive (FP) errors, whereas lower recall results in more false-negative (FN) errors. For a given input gesture and recognition algorithm, there is a trade-off between precision and recall, where the balance between the two types of errors can be adjusted. However, such adjustments cannot eliminate all errors. As a result, prior work to improve recognition accuracy has sought to develop more advanced gesture recognition algorithms [23] or to explore approaches that adjust or modify gesture input languages to better distinguish intentional input from other user behaviors. Examples include introducing "delimiter" gestures that rarely occur naturally and must be performed before an input gesture [12] or choosing a set of input gestures that are unlikely to occur naturally [11, 31]. These approaches can reduce error rate, but also run the risk of making gesture input more time consuming and less intuitive.

Other research has examined the consequences of recognition errors on user experience. Negulescu and Katsuragawa, for example, suggested that FP errors may be more damaging to user experience than FN errors, as the latter can be resolved by simply performing the gesture again [15, 20]. Building on this premise, they developed a bi-level thresholding approach that dynamically adapted a recognizer's threshold to decrease FP errors, while increasing the probability that a repeated gesture will be recognized following a FN error [15, 20]. More recently, Lafreniere et al. established that users are willing to spend more time spent recovering from FN errors if it means that they can avoid FP errors [17], and demonstrated that FP errors can be particularly frustrating due to the attentional costs of noticing and correcting unintended input. In summary, prior work has shown that FP errors are particularly costly for user experience.

Recent research shows that user's implicit goal directed gaze behavior [27] and scan paths [7] can be informative of different motor cognitive states including users' intent [21], task confidence [10], error [28], can be used to provide explanations to the user [32] among other applications.

Peacock et al. demonstrated that gaze behavior following a click event could be used to distinguish FP errors from intended clicks [21]. Participants performed a simple tile selection task in which they searched a grid of tiles for target items, which they were instructed to select. The system would occasionally inject a click (i.e., a simulated input FP) when the user was hovered over a non-target item, selecting it. The user was required to de-select this item before proceeding. A logistic regression model trained on gaze behavior was able to distinguish intentional clicks from these injected clicks (i.e., input FP errors) at above chance as early as 50ms after the input event, with a peak AUC-ROC of .81 at 550 milliseconds after the click event.

Sendhilnathan et al. extended this work, demonstrating that a multi-class deep neural network trained on gaze dynamics could successfully discriminate between intentional clicks, input FP errors, and user errors across three tasks: the tile search task from Peacock et al.; a VR room search task with simulated input FP errors; and a VR dice game controlled by pinch gestures in which input FP errors naturally occurred [28]. This model achieved an AUC-ROC-OVR score of 0.78, demonstrating that the results of [21] generalize both to other tasks and to detecting user errors in addition to input FP errors.

The present research picks up where the work above left off, developing error mediation techniques that could be powered by this gaze-based error detection capability, and investigating them in a controlled user study.

## 2.2 Interactive Mediation Techniques

Error recovery has long been established as an important aspect of usability and an important subject for HCI research. In the early 1980s Shneiderman et al. [29] identified guidelines for error notification methods. In the 1990s Nielsen et al. investigated the prevalence of error recovery (e.g., undo actions, notification) in the usability literature of the time. Also in that decade, van der Meij et al.'s work on usability design principles emphasized error recognition and recovery as a key area [30] , and Abowd and Dix published work diving deep into the user experience of Undo mechanisms and presenting recommendations for their design [1].

In the early HCI work on errors outlined above, the term "error" referred to errors made by the user, but the emergence and integration of AI techniques and speech and handwriting recognizers into interactive systems opened the possibility of the system making errors as well, and with it the need to deal with greater ambiguity. Horvitz's Principles for Mixed-Initiative Interfaces [14] calls out the need to employ dialog to resolve uncertainties about a user's intentions (principle 5), and to design services and alerts to minimize the cost of poor guesses about action and timing (principle 7). Around the same time, Mankoff et al. developed a design space for *mediation techniques*, identifying several types of ambiguity that can occur in recognition-based systems, and broad classes of techniques for resolving this ambiguity, including *repetition* (i.e., the user makes corrections by repeating input) and *choice* (i.e., the user chooses from different possible interpretations of their input, for example from an n-best list) [18]. Prompting the user to choose a correct interpretation is a frequently used approach to disambiguation in input techniques as well. In a recent survey of 3D object selection techniques [3], Argelaguet and Andujar included a category for manual techniques that prompt users to decide among several potential targets, e.g., by cycling through all potential targets [13], displaying targets in a list or menu [9, 16], or by utilizing an additional degree of freedom on an input device [4, 8, 9]. Finally, in commercial software there are individual examples of mediation provided after a system-triggered changes, such as when a word processor automatically corrects the spelling of what it detects to be a mistyped word and prompts the user with a simple interface to undo the auto-correction.

While past work has explored a range of applications of interactive mediation, the idea of employing interactive mediation in response to false positive input errors has received little attention. Schmid et al.'s investigation of input errors caused by "last-instant" changes to system state (e.g., when the user clicks just as the interface updates, causing an unintended item to move under the cursor) proposed the idea of prompting the user with a dialog when these situations are detected, but noted that "careful evaluation should be conducted in order to assess the acceptability of this type of solution." [25]. Other work, such as Schwarz et al.'s interface framework for handling inputs with uncertainty [26] has the potential to support this type of mediation, by enabling a system to represent in parallel multiple potential interpretations of user input, and roll back state for incorrect interpretations when the user's true intention becomes clear, but did not investigate the design of specific error mediation techniques.

The present work fills a gap by investigating error mediation techniques that can be triggered in response to detecting that an input event from a recognition-based input system may have been the result of a false positive. In particular, we investigate the case where such an error may be detected with a short delay after the input event occurs, consistent with the time it would take an error to be recognized through the gaze-based error detection models investigated by Peacock et al. [21] and Sendhilnathan et al. [28].

## 2.3 Highlighting Changes in User Interfaces

A distinguishing feature of the present research over existing work on mediation techniques is that there is a delay of upwards of 550ms between when the error occurs and when the error is detected and mediation is engaged. As we will discuss in the next section, this creates a risk that the user may miss that an error has occurred, or may make it more difficult for the user to identify what was changed as a result of the error.

Past work has recognized the difficulties that can result when changes to an interface are missed by a user and has proposed techniques to address this challenge. For example, Baudisch et al.'s Phosphor technique used an afterglow effect to illustrate changes that occurred to widgets, and Phosphor was found to have performance benefits over animations that replayed widget changes [5]. For changes that occur outside of a user's field of view, Bezerianos et al.'s Mnemonic Rendering proposed an image-based approach that stored a history of pixel-level changes that a user may have missed and then visualized those changes to the user [6]. Our error mediation technique designs draw on some of the ideas investigated

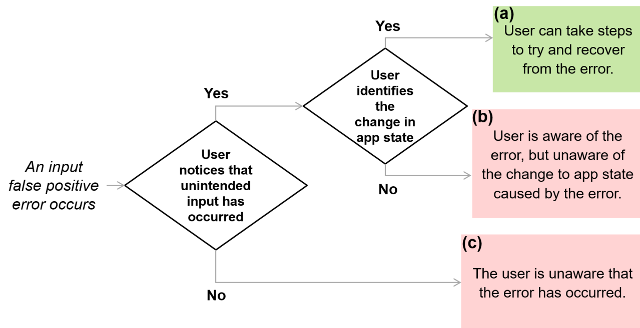



**Figure 2: Three possible outcomes when an input false positive error occurs, from the perspective of the user. The user may (a) notice that input occurred and identify the change to application state; (b) notice that unintended input occurred but fail to identify the resulting change to application state; and (c) fail to notice that unintended input was performed at all. Error mediation techniques must be able to address both (b) and (c).**

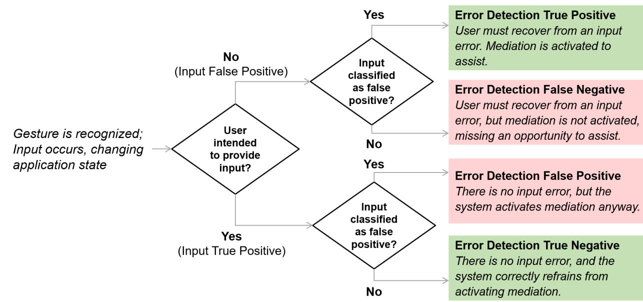



**Figure 3: Flow chart showing four possible scenarios that can occur after an input event, depending on the user's true intention and the error detection model's classification.**

in these past works but for the specific purpose of recovery from recognition errors.

## 3 ERROR MEDIATION DESIGN

As a first step toward designing error mediation techniques, we considered the user experience of input false-positive (*input FP*) errors, and how this might change with the addition of error mediation triggered by an error detection model.

When a user intentionally provides input to a system (i.e., an input true-positive) we can assume that the user has at least some of their attention focused on the system. In contrast, when an input FP error occurs the user may be paying little or no attention to the system. As a result (1) the user may not notice the feedback that is provided by the system to indicate that input has occurred, and (2) the user may not be aware of the change to the application state, if any, that has resulted from the erroneous input. Depending on these two possibilities, there are three potential outcomes outlined in Fig. 2. The optimal case for error recovery is when the user both notices the erroneous input and is able to identify the effects that it has had on the application state (Fig. 2a), but the user may also notice the input but be unaware of what change in application state has occurred as a result of the input (Fig. 2b), or fail to notice that input has occurred at all (Fig. 2c).

Based on the above discussion, we argue that error mediation techniques should fulfill three functional requirements:

**R1. Notification:** Error mediation should assist the user with noticing that unintended input has been provided to the system as a result of an input false positive error.

**R2. Diagnosis:** Error mediation should assist the user with understanding what changes to the application state, if any, have occurred as a result of the unintended input.

**R3. Recovery:** Error mediation should assist the user with reversing the changes to application state, or otherwise recovering from the effects of the error.

In addition to these three functional requirements, error mediation should be designed to be minimally disruptive if it is invoked on an input the user intended to perform. This is a requirement because error detection models such as those proposed by Peacock et al. [21] and Sendhilnathan et al. [28] are often probabilistic and may mis-classify a user's intentional input as an input false positive (i.e., an *error-detection false positive error*), leading to mediation when it is not necessary.

To inform our efforts to develop designs for error mediation techniques, we found it valuable to consider four user experience scenarios that can occur after input is recognized (Fig. 3), depending on the ground truth about a user's intentions (i.e., whether or not the user intended to provide the input to the system), and the error-detection model's classification of the input (i.e., whether it classifies the input event as the result of a false positive or not). Systematically considering the user experience in each of these four scenarios was useful to avoid over-indexing on the case where the error mediation is useful (i.e., an error-detection true positive), and ignoring the cases where it may work against the user's intentions (e.g., an error-detection false positive).

Based on the requirements and considerations defined above, we developed four error mediation techniques for the purpose of eliciting feedback on the overall concept of interactive error mediation, investigating key elements of their design, and testing whether error mediation can help users to recover more quickly from errors.

### 3.1 Error Mediation Designs

Based on the requirements discussed in the previous section, we developed four mediation techniques by varying two design variables. The first variable, *notification method*, defines the means by which the system notifies the user that an error may have occurred (to address requirement R1), and assists the user with understanding the changes to application state that have resulted from the input event (to address R2). In particular, we were interested in this design variable because there is a potential trade-off between notification methods that are difficult for the user to miss (such as in a heads-up display (HUD) presentation) and those that clearly indicate the

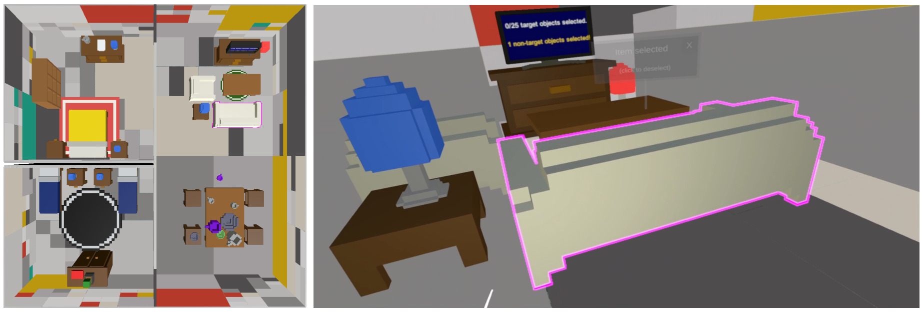



**Figure 4: The study environment layout (left) and the environment from a participant's view (right). The television in the living room showed the number of target and non-target objects selected by the participant. Selected objects were highlighted using a magenta colored outline.**

change to app state (e.g., by spatially locating the notification on the affected object).

The second design variable, *system initiative*, defines how the system enables error recovery (R3). In particular, we were interested in how users would react to the system automatically reversing changes to application state on their behalf, versus facilitating error recovery (e.g., through a prompt) but leaving it to the user to reverse the changes. Understanding the acceptability of automatic recovery is important because automatic recovery could hold the promise of the fastest recovery times, but also creates the possibility of the system sometimes reversing a user's intentional input as a result of a false positive on the error detection model, which could cause frustration.

In the section below we describe each of the four techniques for an example task in which objects in a 3D environment can be selected by pointing at an object and performing a simple pinch gesture. For each of the techniques described below, suppose the user was not intending to select an object, but an input FP occurred, mistakenly selecting one of the objects and triggering the mediation technique.

*3.1.1 Technique 1 (Spatial + Recovery Facilitation):* In Technique 1, upon detecting that an input FP may have occurred, the system presents a notification above the selected object, stating that the item was selected and providing options to undo the selection or dismiss the notification. If the notification is not interacted with for a short time (2 seconds), the dialog begins to fade, and then fades out fully over 10 seconds, after which the dialog dismisses itself. The dialog acts as a notification, identifying the change to application state that resulted from the input, and also helps facilitate recovery if the selection was the result of an input FP (through the 'Undo' button), but allows the user to ignore the dialog and continue working if the selection was intentional (i.e., the case of an error-detection FP).

*3.1.2 Technique 2 (HUD + Recovery Facilitation):* Technique 2 is a variation on Technique 1 in which the notification is presented in a heads-up display (HUD) instead of spatially over the affected object. To indicate the object associated with the notification, an animated arrow is displayed over the object in the scene, which is highlighted when the user hovers over the HUD notification. Otherwise the dialog and behavior is identical to Technique 1.

The rationale behind this design was to make the notification noticeable regardless of where the user is looking, even if the affected object happens to be outside the user's field of view when the input FP is detected.

*3.1.3 Technique 3 (Spatial + Automatic Recovery):* Technique 3 is similar to Technique 1, but with the addition of automatic recovery. Upon detecting that an input FP error may have occurred, the system automatically reverses the effects of the input action and presents a notification above the de-selected object, stating that the item was automatically de-selected and providing options to undo the de-selection or to dismiss the notification. The notification serves to alert the user that the system has de-selected the object on their behalf and provides the user the option to reverse the de-selection. As with the previous techniques, the dialog will fade out and automatically dismiss itself if it is not interacted with, enabling the user to continue their task if the system correctly recovered from the error. In the case of an error-detection FP, the user must click 'Undo' or re-select the object to reverse the erroneous "recovery" by the system.

*3.1.4 Technique 4 (HUD + Automatic Recovery):* Technique 4 is identical to Technique 3 but with a HUD rather than spatial presentation. As with Technique 2, to indicate the object associated with the HUD notification, an animated arrow is displayed over the affected object in the scene which is highlighted when the user hovers over the notification.

The next section describes a study that was conducted to investigate the error mediation techniques presented above.

# 4 STUDY

The objective of this study was to investigate whether error mediation techniques can save time and effort when recovering from input false-positive errors, and to gain insights into the error mediation technique designs introduced in section 3.1.

## 4.1 Study Environment

A virtual environment was developed in which a user could move about freely, and select (or de-select) objects by pointing and clicking with a ray cursor. The ray cursor was used in place of gesture interactions that typically rely on recognition models with a fixed and non-controllable error rate. The study environment contained three rooms — a living room and two bedrooms — populated with

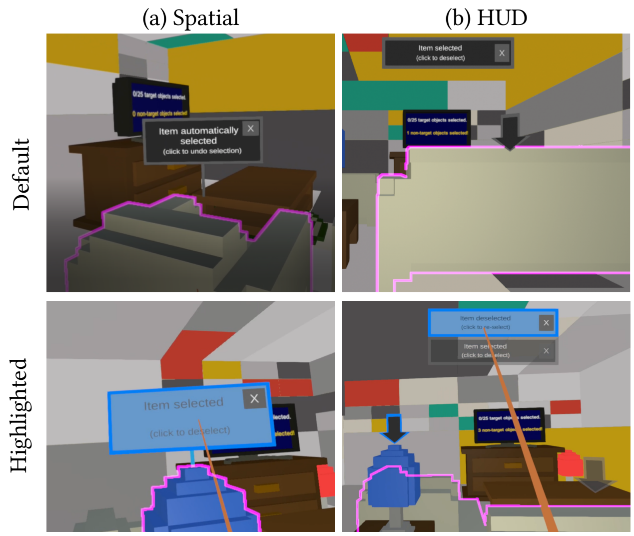



**Figure 5: Participants were notified of detected errors with a prompt (a) spatially attached to the object where the error occurred, or (b) in the heads-up display (HUD) with an arrow highlighting the affected object. When participants hovered over the notifications, they become opaque and were highlighted with a blue outline. If an arrow was associated with the notification, it was also highlighted.**

selectable objects, such as furniture, lamps, books, and panels covering the walls, ceilings, and floors (Fig.4). The intent of this environment was to simulate a real-world space augmented with AR features.

## 4.2 Input Method

An HTC VIVE controller held in the participant's dominant hand controlled a ray cursor, indicated by an orange line (shown in the lower right image of Fig. 5). The trigger button on the controller selected or deselected the object hovered over with the pointer. When such a click action was performed, a click sound was played and the pointer would briefly become wider and change from orange to yellow. Selected objects were indicated using a magenta-colored outline.

While we are ultimately interested in mediation for gesture-based input, it is difficult to control and measure error rates when using a live gesture recognizer. Thus, we follow past work and use a VR controller with a physical button, enabling us to inject clicks to simulate false positive errors at a controlled rate [17, 21, 28].

A second HTC VIVE controller, held in the participant's non-dominant hand, was used to navigate the environment using a standard teleportation-based navigation method. The user pointed to a desired location with a arc-shaped pointer and pressed the trigger button to teleport to the indicated location.

## 4.3 Error Injection Approach

To simulate input false-positive errors, unintended clicks were injected on a random schedule while the participant's cursor hovered over selectable objects in the environment. These clicks could select or de-select objects, and were identical to user-initiated clicks in terms of visual and audio feedback. At the start of each block, a random schedule for click injection times was created, sampling 20 random times over a 5-minute period such that individual injection times were at least 5 seconds apart. At any time where the user was hovering over an object, the timer would run, and a click would be injected when it hit one of the pre-generated injection times.

A potential challenge with the above approach is that participants might avoid injected input FP errors by intentionally avoiding hovering over objects that they do not want to select. To address this, the walls, ceilings, and floors of the environment were tiled with selectable panels, such that no matter where the participant's controller was pointing, the ray cursor landed on a selectable object.

## 4.4 Study Task

Participants were asked to select 25 objects (i.e. items in the environment, panels on the walls, furniture) of a specified target color. The environment was initialized with at least 26 objects of the target color, to make the task slightly easier and avoid the case where a user struggles to find all items of the target color. The colors of furniture and other items in the environment were fixed, but the color of panels on the walls was dynamically set. Initially, each panel in the environment was randomly assigned either one of the possible target colors (red, yellow, or green) with a probability of 0.2, or a neutral color from a set of 6 neutral colors. Next, neutral-colored panels were randomly recolored to the target color until the correct number of target items was reached.

The non-panel objects in the scene varied slightly in color from object to object, and in pilots we observed that participants were sometimes confused as to whether an object "counted" as a given color. To provide a general progress indicator and assist with this challenge, a text summary of the current number of selected target and non-target objects was displayed in the environment, attached to the television 3D object (Fig. 4).

## 4.5 Apparatus

Participants completed the study wearing an HTC VIVE Pro Eye head-mounted display connected to an MSI GS66 laptop running Windows 10. The study system was built using Unity 2021.1.14.

## 4.6 Study Design and Procedure

The study followed a mixed design with within-subjects factor *Mediation Technique* and between-subjects factor *Error Detection Model*. The Mediation Technique factor had four levels, corresponding to the four techniques introduced in Section 3.1 and shown in Fig. 5. Participants completed one block with each of the four mediation techniques, with the order counterbalanced across participants using a balanced Latin square.

To investigate the relationship between the accuracy of the error detection model and response to the mediation techniques, the Error Detection Model factor had four levels, simulating four combinations of true-positive (TPR) and false-positive (FPR) rates for an error detection model. The simulated error detection model determined when an injected click would be "detected" by the model. For example, for a model with TPR=90% and FPR=5%, there was a 90% chance that the error mediation would be activated after an injected click, and a 5% chance it would be activated after a user-initiated click. When activated, the error mediation techniques

**Table 1: Participant demographics for the baseline and four simulated error model conditions.**

| | Model 1 (*P1, P2, P3, P4*) | Model 2 (*P5, P6, P7, P8*) | Model 3 (*P9, P10, P11, P12*) | Model 4 (*P13, P14, P15, P16*) | Without mediation (N=19) |
|---|---|---|---|---|---|
| Age | 30, 65, 62, 27 | 36, 45, 23, 28 | 63, 55, 22, 41 | 45, 25, 29, 38 | M=37.53 (SD=14.02, range: 22 to 65) |
| Gender | 4 Female | 1 Female, 3 Male | 2 Female, 2 Male | 1 Female, 3 Male | 8 Female, 9 Male, 1 Non-binary, 1 N/S |
| Handedness | 4 Right | 4 Right | 4 Right | 4 Right | 19 Right |

would be displayed 350ms after the associated input event, a delay chosen based on the lens sizes for input FP detection in Peacock et al. [21]. The specific models included in the study were: a high TPR and low FPR (Model 1: TPR=90%, FPR=5%), a high TPR and high FPR (Model 2: TPR=90%, FPR=20%), a low TPR and low FPR (Model 3: TPR=70%, FPR=5%), and a low TPR and high FPR (Model 4: TPR=70%, FPR=20%). Each participant was assigned one of the four simulated error detection models.

Prior to the first study block, participants completed a short training session to familiarize themselves with the system's controls. The participant was asked to select a cube, and then de-select a cylinder. Next, an unintended click was injected at the first panel the participant hovered their pointer over, and they were prompted: "An unintended click occurred. Fix it by de-selecting/re-selecting the object". Once this was done, the participant was asked to teleport to the second room and select a sphere, after which the training session ended.

Next, participants completed 4 study blocks. Before the start of each block, the participant was prompted with the target color of object to select for that task, and a short description of the mediation technique that would be active in the block. The participant was then placed in the virtual environment at a fixed starting location and orientation. The block concluded once the participant had selected at least 25 objects of the prompted color and no objects of other colors. This criteria was designed to force participants to deselect non-target items, but leave it up to them to decide whether to do so immediately or not.

After each block, participants completed a post-block questionnaire that asked them to indicate their willingness to continue using that block's mediation technique by answering the question "If you had to do this block again, and you could choose between continuing to use this mediation technique, or disabling all mediation techniques, what would you choose?" At the end of the study, participants completed a short post-study questionnaire which included qualitative feedback questions on aspects of the mediation techniques to understand preference for Spatial versus HUD presentation, and automatic correction versus recovery facilitation. Demographic information was also provided through the post-study questionnaire. In all, the study took approximately 60 – 75 minutes to complete.

In addition to the study design outlined above, an additional set of participants were recruited to complete the study task without any mediation techniques present, to provide a baseline of error recovery time without error mediation. Apart from having no mediation techniques, these participants experienced the same task, environment, apparatus, and error injection approach.

## 4.7 Participants

Sixteen participants were recruited to complete the study with mediation techniques present (Table 1). An additional set of participants was recruited for a baseline (without mediation) condition, of which 19 were included in final data analysis after filtering out sessions with data logging issues (Table 1). All participants reported having normal vision without the need for corrective lenses. Participants provided written consent to participate and the study was approved by our Institutional Review Board.

# 5 RESULTS

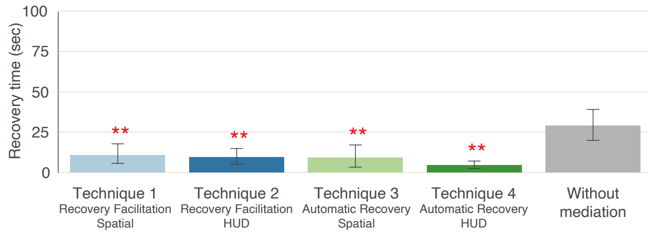
## 5.1 Error Recovery Time

*5.1.1 Recovery Time by Mediation Technique.* In order to evaluate the effectiveness of error mediation techniques, we analyzed the impact of error mediation techniques on both the i) time taken by users to recover from errors (urgency of error recovery) and ii) prioritization of error recovery, relative to a baseline condition without any mediation.

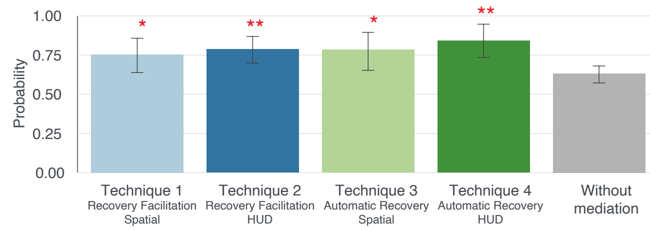
*5.1.2 Error mediation increased the urgency of error correction.* We defined error recovery time as the duration between the moment an injected click resulted in the selection of a non-target item (or deselection of a target item) and the moment the error was rectified by reversing the selection (or deselection). It is important to note that this duration includes cases where an injected click led to an error, but the mediation technique was not triggered (i.e., false negatives in error detection), as well as instances where the error was corrected either by utilizing a mediation technique or directly clicking on the object. As a result, in the analyses that follow, the error recovery times for automatic techniques may be greater than zero.

All four error mediation techniques showed a statistically significant decrease in average error recovery time compared to the baseline condition (i.e., without error mediation) (Fig. 6a; ANOVA, $F(4, 71)=6.60$, $p < .001$). This implies that users recovered significantly faster from errors when an error mediation technique was available in the system, compared to the baseline with no mediation ($M = 29.14 \pm 10.14$): Spatial + Recovery Facilitation ($M = 10.72 \pm 6.41$, $t = 2.89, p < .01$), Spatial + Automatic Recovery ($M = 9.16 \pm 7.23$, $t = 2.89, p < .01$), HUD + Recovery Facilitation ($M = 9.52 \pm 5.48$, $t = 3.16, p < .01$), HUD + Automatic Recovery ($M = 4.48 \pm 1.99$, $t = 3.57, p < .01$). It's notable that these differences are not small – mediation enabled recovery 2.7x to 6.5x faster than without mediation, depending on the technique.

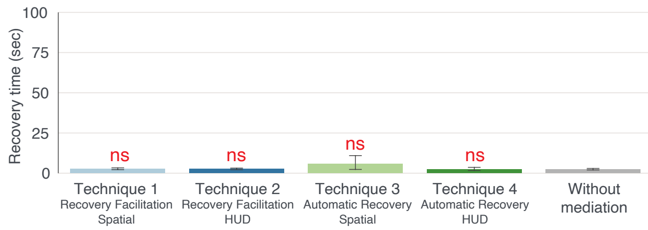
While mediation enabled faster error recovery as compared to the baseline, we did not see a significant difference between the mediation techniques. A two-way ANOVA to analyze the effects of mediation technique and simulated error detection model on error



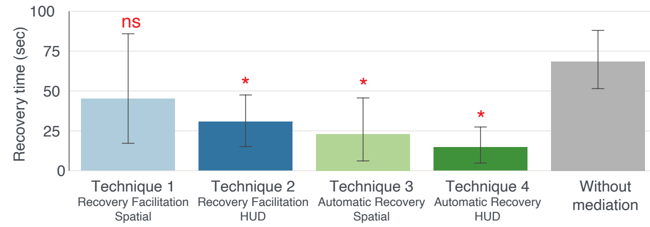

**(a) Average error recovery time.**




**(b) Average proportion of errors recovered from immediately.**




**(c) Average error recovery time (immediate recovery errors only).**




**(d) Average error recovery time (delayed recovery errors only).**

**Figure 6: Error recovery across error detection models. Error bars indicate 95% confidence intervals. The significance markers (*p < .05, **p < .01) denote the results of comparing recovery with the mediation techniques and the baseline condition (without mediation).**

recovery time found no significant effect of mediation technique (F(3, 41)=403.12, $p = 0.42$), no significant effect of the simulated error detection model (F(3, 41)=916.23, $p = 0.11$), and no significant interaction between the mediation technique and the simulated error detection model (F(9, 41)=787.24, $p = 0.77$).

*5.1.3 Error mediation increased the prioritization of error correction.* Next we investigated the extent to which users prioritized error correction in the conditions with error mediation techniques versus without them. To do this, we categorized error recovery instances as "immediate" if the user did not interact with any objects other than the one affected by the error between the time of error occurrence and recovery. Conversely, any instance where the user did interact with other objects during this period was classified as "delayed." This approach enabled us to examine the extent to which mediation techniques encouraged users to identify and correct errors immediately.

The average proportion of errors recovered from immediately with error mediation techniques was significantly higher than without error mediation (Fig. 6b; ANOVA, F(4, 71)=3.49, $p < .05$). Independent samples t-tests showed significantly higher proportions of errors were recovered from immediately for each of the four mediation techniques, as compared to the baseline ($M = 0.63 \pm 0.06$): Spatial + Recovery Facilitation ($M = 0.75 \pm 0.11$, $t = -2.06, p < .05$), Spatial + Automatic Recovery ($M = 0.78 \pm 0.12$, $t = -2.34, p < .05$), HUD + Recovery Facilitation ($M = 0.79 \pm 0.12$, $t = -3.08, p < .01$), and HUD + Automatic Recovery ($M = 0.84 \pm 0.09$, $t = -3.63, p < .01$).

It is not surprising that the automatic recovery techniques resulted in more immediate error recovery, since these techniques corrected the error for the user. However, that this was also the case for the recovery facilitation techniques suggests that one of the ways that mediation can provide benefit is by notifying the user that an error has occurred, increasing the chance that the user corrects the error immediately.

As shown earlier, the use of error mediation techniques significantly promotes prioritization of error recovery. The time taken to recover from an error that has already been prioritized for immediate correction does not vary significantly when mediation techniques are employed compared to when they are not (Fig. 6c).

In contrast, error mediation appears to show a large advantage in situations where there is a delay in error recovery. In cases where there is a delay in error recovery, the use of error mediation techniques results in significantly faster recovery for users than when such techniques are not utilized ($p < .05$, ANOVA; Fig. 6d). Individually, we found significantly faster average recovery times versus the baseline for all but the Spatial + Recovery Facilitation technique: Baseline ($M = 68.48 \pm 19.02$), Spatial + Automatic Recovery ($M = 22.9 \pm 14.75$, $t = 2.75, p < 0.05$), HUD + Recovery Facilitation ($M = 30.67 \pm 15.41$, $t = 2.74, p < 0.05$), HUD + Automatic Recovery ($M = 14.72 \pm 7.22$, $t = 2.76, p < 0.05$), Spatial + Recovery Facilitation ($M = 45.1 \pm 31.07$, $t = 1.25, p = 0.22$). For the recovery facilitation techniques, this may be a result of the techniques notifying the user of errors, which they choose not to recover from until after interacting with other objects. For the automatic recovery techniques, this may reflect cases where the technique activated after the user has interacted with another object, causing the error recovery to be classified as delayed.

In summary, error mediation offers three primary advantages: i) it enables users to recover from errors much more rapidly than they would without mediation, ii) it increase the probability of immediate recovery of errors, and iii) even when there is a delay in the recovery process due to other intervening actions, users are still able to recover more quickly when error mediation techniques are utilized.

*5.1.4 Notification Presentation and System Initiative.* To analyze how error recovery time was influenced by the relative differences in design between the mediation techniques, a RM-ANOVA was performed with factors notification method (HUD vs. Spatial)

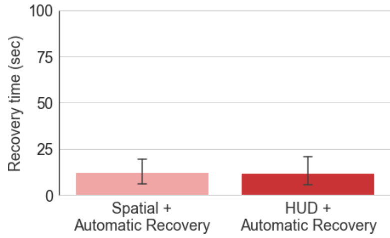



**Figure 7: Average time to recover from error-detection FP errors, for the two automatic recovery techniques.**

and system initiative (Recovery Facilitation vs. Automatic Recovery). We found no significant effect for either notification method (F(1, 15)=1.67, $p = 0.22$), system initiative (F(1, 15=2.55, $p = 0.13$), or the interaction between the two factors (F(1, 15)=0.42, $p = 0.53$). This suggests that there was no relative advantage in terms of improving error recovery time between the designs. In concert with the analysis versus the baseline above, this may suggest that all of the mediation techniques provided an advantage in terms of error recovery time, but do so in different ways. As well, later in the paper we report post-block preferences and qualitative feedback, which shows that participants did have a preference for some designs over others.

*5.1.5 Error-Detection False Positive Recovery.* The analyses in this section so far have focused on the time to recover from input FP errors, but the automatic recovery techniques are also susceptible to error-detection false positives, in which the system incorrectly classifies an intended click as the result of an input FP, triggering automatic correction when it is not required. Fig. 7 shows the average amount of time it took participants to recover from these errors for the two Automatic Recovery techniques. Across all simulated error-detection models, the average recovery time for these errors was 11.9 ± 7.13 seconds for the Spatial technique, and 11.46 ± 9.02 seconds for the HUD. We note that these average recovery times are close to the average error recovery times for the mediation techniques (as opposed to the no-mediation baseline).

*5.1.6 Summary.* Our analysis of error recovery time shows that error mediation can enable faster error recovery from input false positive errors, and that this benefit comes from the techniques' ability to ensure that these errors are corrected more quickly – by notifying the user that the error has occurred in the case of recovery facilitation techniques, and by taking initiative and correcting the error in the case of the automatic recovery techniques. While we saw no difference in recovery time between the techniques, the automatic recovery techniques imposed additional time to recover from error-detection false positive errors.

In the sections that follow we examine post-block responses and qualitative feedback, to gain further insights into the user experience and relative benefits of the different error mediation designs.

## 5.2 Post-Block Responses

In the post-block surveys, participants indicated whether they wished to continue using the associated mediation technique, if they had to do the block again. Fig. 8 shows responses for each technique, and the same data grouped by notification method and level

| | Technique 1 (Spatial + Recovery Facilitation) | Technique 2 (HUD + Recovery Facilitation) | Technique 3 (Spatial + Automatic Recovery) | Technique 4 (HUD + Automatic Recovery) |
|---|---|---|---|---|
| **Model 1** (TPR=90%, FPR=5%) | 4/4 | 1/4 | 2/4 | 3/4 |
| **Model 2** (TPR=90%, FPR=20%) | 2/4 | 2/4 | 2/4 | 1/4 |
| **Model 3** (TPR=70%, FPR=5%) | 2/4 | 1/4 | 3/4 | 2/4 |
| **Model 4** (TPR=70%, FPR=20%) | 2/4 | 2/4 | 2/4 | 2/4 |

| | Grouped by Notification Method | | Grouped by System Initiative Level | |
|---|---|---|---|---|
| | **Spatial** | **HUD** | **Recovery Facilitation** | **Automatic Recovery** |
| **Model 1** (TPR=90%, FPR=5%) | 6/8 | 4/8 | 5/8 | 5/8 |
| **Model 2** (TPR=90%, FPR=20%) | 4/8 | 3/8 | 4/8 | 3/8 |
| **Model 3** (TPR=70%, FPR=5%) | 5/8 | 3/8 | 3/8 | 5/8 |
| **Model 4** (TPR=70%, FPR=20%) | 4/8 | 4/8 | 4/8 | 4/8 |

**Figure 8: The number of blocks for which participants indicated that they would continue using the mediation technique if they were to perform the block again, for each error model. Top: all techniques. Bottom: Grouped by notification method (left) and system initiative level (right).**

of system initiative. For the most accurate simulated error-detection model (Model 1), a majority of participants chose to continue using the mediation techniques (10/16 blocks or 62.5%). Models 2, 3, and 4 saw less willingness to continue using the mediation techniques, with 7/16, 8/16, and 8/16 respectively, which may be explained by the lower accuracy of the simulated models.

Examining the data grouped by notification method (Fig. 8 lower left) shows more support for Spatial (19/32) or 59% of blocks, over HUD (14/32) or 43% of blocks. Participants' qualitative comments, discussed in the next section, were consistent with this trend toward higher support for Spatial notifications.

Examining the data grouped by automatic recovery versus recovery facilitation (Fig. 8 lower right) shows no clear trend, though support for automatic recovery appears to be slightly higher for the low FPR models (1 and 3), which would make sense as error-detection false positives are potentially more damaging for these techniques.

## 5.3 Qualitative Feedback

Post-study questionnaire responses were analyzed for common themes in how participants responded to the two different placements of information (Spatial versus HUD), and whether it was acceptable for the system to take initiative in correcting detected errors.

*5.3.1 Spatial vs. HUD Presentation.* In the post-study questionnaire, participants were asked which notification placement they preferred. In response, they showed a clear preference for the Spatial presentation, with 13/16 participants indicating Spatial and only 3/16 indicating HUD. Participants expressed that the Spatial notifications made it easier to identify and locate the object in the scene, as in the following quote:

> *The notification being attached to the item made it easier to identify which item was in question. -P5*

Participants also expressed that the Spatial presentation assisted with error correction because the notification served as a larger target than the affected object, making the target easier to select.

Conversely, participants expressed that the HUD notifications could be confusing when multiple notifications were present, as it was not immediately clear which object was related to which notification:

> *Spatial placement made it easier to tell which tile the notification was for...I did not like the heads-up display because it did not do a good job of differentiating between selected tiles and the tile the [notification] is actually for. -P11*

Participants also expressed that the HUD notifications could be difficult to view and target, could be "intrusive", and at times occluded objects in the scene.

Collectively, the above feedback suggests a clear preference for the Spatial notifications because of their ability to immediately indicate the affected item and offer a target that is easier to acquire than the affected object itself, while being less intrusive if the user wishes to ignore them.

*5.3.2 Automatic Recovery.* In response to a post-study questionnaire question about whether they liked the automatic correction behavior used by some of the mediation techniques, 9/16 participants responded that they liked the behavior, while 6/16 responded that they did not. The most commonly cited benefit of automatic recovery was time savings, as in the following quote:

> *It was helpful that it automatically fixes when you select or deselect an item by mistake, since it saves you time. -P12*

While the time savings of automatic recovery was appreciated, some of the participants qualified this benefit, referencing that the technique at times worked against their intentions:

> *The mediation at times automatically turned off [a] correct selection. When it worked correctly it was helpful and did save me time. -P2*

Several participants mentioned instances where error detection FP or FN errors had occurred, or expressed a general dislike for the system acting incorrectly on their behalf (e.g., P11: "I didn't like this technique as much because it deselected one of my red squares unintentionally", P15: "I hate that it kept selecting things I didn't want to", P6: "It caught me off guard. It was faster and easier to fix myself").

Overall, the above feedback suggests that caution should be taken in employing automatic recovery methods, and that acceptance of this approach may be tied to the false positive rate on the input FP detection model. While they were appreciated for the time savings they can provide, it is clear that instances where the system incorrectly takes action can be frustrating for users. The above being said, we found it encouraging that participants' main issue with this feature was that it at times made the wrong decision, rather than a more fundamental criticism of the feature taking initiative to change application state on the user's behalf.

## 6 DISCUSSION

Overall, our study findings indicate that error mediation techniques can significantly reduce the time to recover from input false positive errors, with average error recovery times 2.5x to 6.5x faster than without error mediation, and that this benefit comes from increasing the proportion of errors that are corrected immediately, suggesting that error mediation helps users to avoid overlooking errors when they occur. This section discusses our results in greater detail, and provides guidance to designers and researchers interested in building on these results.

The above findings are encouraging for the idea of adding error-awareness to input systems. In particular, they suggest that the capability to detect input false-positive errors after they have occurred based on behavioral signals, such as the eye-gaze models developed in recent work [21, 28], can directly benefit users by powering interactive error mediation techniques. Moreover, our finding that error mediation techniques can help users to notice input false positive errors is consistent with recent work by Lafreniere et al. [17], which indicates that these errors are particularly challenging because of the attentional cost to notice them and diagnose their effects when they occur. Our results suggest that error mediation can lessen this burden on the user.

Finally, it is important to emphasize that the benefits of error mediation are entirely compatible with existing approaches to improving input recognition. In an ideal system, eye-gaze and other behavioral signals preceding an input event can be used to improve recognition accuracy, and the same sets of signals can be used to detect input errors that "slip through the cracks", enabling the system to employ error mediation and assist the user with error recovery. Ultimately this work contributes to advancing a vision of a comprehensive approach to addressing recognition errors in input systems through both error reduction and mitigation.

### 6.1 Error Mediation Design – Spatial vs. HUD

Our study results indicate strong support for a spatially-locked presentation of error mediation. In particular, the spatial presentation used in two of our techniques was appreciated for its ability to help the user quickly locate the affected object in the scene, and to assist with correcting the error by providing a larger target. In contrast, the HUD presentation's indirect approach to referencing objects in the scene created several challenges. The promise of the HUD presentation was that notifications could not be missed by the user, but our results suggest that this design imposed an additional "de-referencing" cost on the user to locate the associated object in the scene. Displaying the notification in the user's view also created a difficult trade-off between providing a noticeable and easy notification to select while not occluding content in the scene or creating visual distraction.

The above results on presentation design suggest that a spatially-locked presentation of error mediation has a benefit for addressing the challenges of noticing and diagnosing input false positive errors raised in prior work [17]. A takeaway for designers is to consider the additional cognitive costs imposed on the user by the error mediation design, such as the need for de-referencing to locate an affected object. Future work could investigate different methods of presenting HUD notifications, or ways of supporting de-referencing to locate an affected object. It would also be interesting to consider hybrid techniques – for example, a spatial notification when an affected object is within the user's field of view, and a HUD notification when it is outside of the user's view, to direct the user's attention to the affected object.

## 6.2 Error Mediation Design – System Initiative

Our study results showed mixed support for techniques that take initiative to correct errors on the user's behalf. While a slight majority of participants indicated support for the automatic correction behavior, participants also expressed frustration that the automatic recovery techniques could act counter to their intentions. At the same time, participants expressed support for the ability of automatic correction to save time, and did not object to the principle of the system taking initiative to correct errors on their behalf. Overall, these results suggest that automatic correction could be a viable approach to error mediation, but that its success is likely to depend on the accuracy of the error detection model – particularly its false positive rate. Further studies are required to investigate how acceptance of automatic correction varies with the accuracy characteristics of an error detection model.

While support for automatic correction was mixed, a positive finding is that the recovery facilitation techniques achieved error recovery times that are comparable to that of automatic recovery, and significantly faster than without mediation. This shows that error mediation can provide benefits even without taking initiative to correct errors on behalf of the user.

## 6.3 Future Work and Limitations

We see several interesting directions for future work. First, it would be valuable to test error mediation in an end-to-end system, triggered by a real-time error detection model. Additional challenges may arise when integrating the mediation techniques with a real-time model. For example, prior work on using a real-time gaze model to trigger navigation aids noted the challenge of setting the model's prediction threshold, prompting the authors to add a feature that would allow the user to dynamically adjust this threshold [2]. As well as surfacing such challenges, an end-to-end system would enable the investigation of dynamic approaches to the trade-off between activating error mediation sooner after an input event, with less certainty that an input FP error has occurred, versus activating later with greater certainty.

Second, the number of participants in the present study was small for understanding how recognizer accuracy drove acceptance of the error mediation techniques. Moreover, though we observed significantly faster error recovery times for the mediation technique conditions over the baseline, the unequal sample sizes between these conditions may have affected the validity of our analysis. A follow-up study with more participants could provide deeper insights on the relationship between the error-detection model's performance and the effectiveness, and acceptability, of error mediation. It would also be interesting to run studies over a longer period of time, to understand how users perceptions and use of error mediation techniques changes with greater experience and familiarity using them.

Finally, we conducted our exploration with the assumption that recovering from errors immediately is desirable for users. In many cases we believe this will be true, since with a greater delay the user may have trouble recalling whether a change was desired or not, and some mechanisms for reversing changes in application state (like application-wide undo) become more technically complicated once further changes are performed. That being said, it would be interesting to investigate how the mediation designs presented here could be modified to enable users to defer error recovery until a later time, e.g., by persisting notifications for later review. This could have potential advantages, such as allowing the user to stay in the flow of work and then fixing errors at a later time.

# 7 CONCLUSION

In order to realize a vision of seamlessly integrating interactions with digital content amongst our everyday activities, there is a need for systems that can reliably distinguish intentional input from other user behaviors. Motivated by recent results showing that eye-gaze behavior can be used to detect input errors in the moments after they occur, this research has contributed a first exploration of interactive error mediation techniques that could be powered by such a capability, and a first empirical study showing that these techniques can provide benefits to users. These results move us closer to a vision of error-aware input systems with the ability to dynamically assist the user with recovering from input errors, creating a more natural user experience.

# REFERENCES


[1] Gregory D Abowd and Alan J Dix. 1992. Giving undo attention. *Interacting with Computers* 4, 3 (1992), 317–342. https://doi.org/10.1016/0953-5438(92)90021-7

[2] Rawan Alghofaili, Yasuhito Sawahata, Haikun Huang, Hsueh-Cheng Wang, Takaaki Shiratori, and Lap-Fai Yu. 2019. Lost in Style: Gaze-driven Adaptive Aid for VR Navigation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 348, 12 pages. https://doi.org/10.1145/3290605.3300578

[3] Ferran Argelaguet Sanz and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers and Graphics* 37, 3 (May 2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[4] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting. 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, 1–12. https://doi.org/10.1145/3290605.3300331

[5] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 169–178. https://doi.org/10.1145/1166253.1166280

[6] Anastasia Bezerianos. 2006. Mnemonic Rendering: An Image-Based Approach for Exposing Hidden Changes in Dynamic Displays. In *Proc. UIST2006*. 159–168.

[7] Charlie S Burlingham, Naveen Sendhilnathan, Oleg Komogortsev, T Scott Murdison, and Michael J Proulx. 2024. Motor "laziness" constrains fixation selection in real-world tasks. *Proceedings of the National Academy of Sciences* 121, 12 (2024), e2302239121.

[8] William Delamare, Céline Coutrix, and Laurence Nigay. 2013. Mobile Pointing Task in the Physical World: Balancing Focus and Performance While Disambiguating. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 89–98. https://doi.org/10.1145/2493190.2493232

[9] Tovi Grossman and Ravin Balakrishnan. 2006. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1166253.1166257

[10] Aakar Gupta, Naveen Sendhilnathan, Jess Hartcher-O'Brien, Evan Pezent, Hrvoje Benko, and Tanya R Jonker. 2023. Investigating Eyes-away Mid-air Typing in Virtual Reality using Squeeze haptics-based Postural Reinforcement. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–11.

[11] Seongkook Heo, Jiseong Gu, and Geehyuk Lee. 2014. Expanding Touch Input Vocabulary by Using Consecutive Distant Taps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2597–2606. https://doi.org/10.1145/2556288.2557234

[12] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in

Scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 451–460. https://doi.org/10.1145/1054972.1055035

[13] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A Survey of Design Issues in Spatial Input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (Marina del Rey, California, USA) *(UIST '94)*. Association for Computing Machinery, New York, NY, USA, 213–222. https://doi.org/10.1145/192426.192501

[14] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[15] Keiko Katsuragawa, Ankit Kamal, Qi Feng Liu, Matei Negulescu, and Edward Lank. 2019. Bi-Level Thresholding: Analyzing the Effect of Repeated Errors in Gesture Input. *ACM Trans. Interact. Intell. Syst.* 9, 2–3, Article 15 (apr 2019), 30 pages. https://doi.org/10.1145/3181672

[16] Regis Kopper, Felipe Bacim, and Doug A. Bowman. 2011. Rapid and accurate 3D selection by progressive refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 67–74. https://doi.org/10.1109/3DUI.2011.5759219

[17] Ben Lafreniere, Tanya R Jonker, Stephanie Santosa, Mark Parent, Michael Glueck, Tovi Grossman, Hrvoje Benko, and Daniel Wigdor. 2021. False Positives vs. False Negatives: The effects of recovery time and cognitive costs on input error preference. *34th ACM Symposium on User Interface Software and Technology*.

[18] Jennifer Mankoff, Scott E. Hudson, and Gregory D. Abowd. 2000. Providing Integrated Toolkit-Level Support for Ambiguity in Recognition-Based Interfaces. 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, 368–375. https://doi.org/10.1145/332040.332459

[19] George Nagy, Thomas A. Nartker, and Stephen V. Rice. 1999. Optical character recognition: an illustrated guide to the frontier. In *Electronic Imaging*.

[20] Matei Negulescu, Jaime Ruiz, and Edward Lank. 2012. A Recognition Safety Net: Bi-Level Threshold Recognition for Mobile Motion Gestures. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 147–150. https://doi.org/10.1145/2371574.2371598

[21] Candace E. Peacock, Ben Lafreniere, Ting Zhang, Stephanie Santosa, Hrvoje Benko, and Tanya R. Jonker. 2022. Gaze as an Indicator of Input Recognition Errors. In *ACM Symposium on Eye Tracking Research and Applications*. 14 pages.

[22] Gerhard Rigoll, Andreas Kosmala, and Stefan Eickeler. 1997. High Performance Real-Time Gesture Recognition Using Hidden Markov Models. In *Gesture Workshop*.

[23] Dean Rubine. 1991. Specifying Gestures by Example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. Association for Computing Machinery, New York, NY, USA, 329–337. https://doi.org/10.1145/122718.122753

[24] Jaime Ruiz and Yang Li. 2011. DoubleFlip: A Motion Gesture Delimiter for Mobile Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2717–2720. https://doi.org/10.1145/1978942.1979341

[25] Philippe Schmid, Sylvain Malacria, Andy Cockburn, and Mathieu Nancel. 2020. Interaction Interferences: Implications of Last-Instant System State Changes. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 516–528. https://doi.org/10.1145/3379337.3415883

[26] Julia Schwarz, Scott Hudson, Jennifer Mankoff, and Andrew D. Wilson. 2010. A Framework for Robust and Flexible Handling of Inputs with Uncertainty. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 47–56. https://doi.org/10.1145/1866029.1866039

[27] Naveen Sendhilnathan, Debaleena Basu, Michael E Goldberg, Jeffrey D Schall, and Aditya Murthy. 2021. Neural correlates of goal-directed and non–goal-directed movements. *Proceedings of the National Academy of Sciences* 118, 6 (2021), e2006372118.

[28] Naveen Sendhilnathan, Ting Zhang, Ben Lafreniere, Tovi Grossman, and Tanya R. Jonker. 2022. Detecting Input Recognition Errors and User Errors Using Gaze Dynamics in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 38, 19 pages. https://doi.org/10.1145/3526113.3545628

[29] Ben Shneiderman. 1982. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology* 1, 3 (1982), 237–256. https://doi.org/10.1080/01449298208914450 arXiv:https://doi.org/10.1080/01449298208914450

[30] Hans van der Meij and John M Carroll. 1995. Principles and Heuristics for Designing Minimalist Instruction. *Technical Communication: Journal of the Society for Technical Communication* 42, 2 (1995), 243–61.

[31] Bryan Wang and Tovi Grossman. 2020. *BlyncSync: Enabling Multimodal Smartwatch Gestures with Synchronous Touch and Blink*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376132

[32] Mengjie Yu, Dustin Harris, Ian Jones, Ting Zhang, Yue Liu, Naveen Sendhilnathan, Narine Kokhlikyan, Fulton Wang, Co Tran, Jordan L Livingston, et al. 2024. Explainable Interfaces for Rapid Gaze-Based Interactions in Mixed Reality. *arXiv preprint arXiv:2404.13777* (2024).