Mise-Unseen: Using Eye-Tracking to Hide Virtual Reality Scene Changes in Plain Sight

Sebastian Marwecki\textsuperscript{1,2}, Andrew D. Wilson\textsuperscript{1}, Eyal Ofek\textsuperscript{1}, Mar Gonzalez Franco\textsuperscript{1}, Christian Holz\textsuperscript{1}

\textsuperscript{1}Microsoft Research
Redmond, WA, USA

\textsuperscript{2}Hasso Plattner Institute,
University of Potsdam, Germany

![Figure 1](image)

**Figure 1**: (a) Mise-Unseen uses eye-tracking in VR headsets to hide changes that occur inside the user’s field of view by applying them outside the user’s fovea. Mise-Unseen unnoticeably changes the scene as the user is focusing elsewhere (circle): (b) cross-facing pieces together (arrow) to help the user solve this puzzle, (c) swapping the gallery painting to adapt to the user’s detected interest in modern art, (d) hiding the low fidelity of this “explosion”, (e) matching the virtual axe’s position to the haptic prop following the detected user interest, (f) shifting storage racks while walking to adapt to a lack of physical space, (g) reducing motion sickness during teleportation by blending static images outside the fovea, and (h) updating hints to prevent the user from solving this riddle too quickly.

**ABSTRACT**
Creating or arranging objects at runtime is needed in many virtual reality applications, but such changes are noticed when they occur inside the user’s field of view. We present Mise-Unseen, a software system that applies such scene changes covertly inside the user’s field of view. Mise-Unseen leverages gaze tracking to create models of user attention, intention, and spatial memory to determine if and when to inject a change. We present seven applications of Mise-Unseen to unnoticeably modify the scene within view (i) to hide that task difficulty is adapted to the user, (ii) to adapt the experience to the user’s preferences, (iii) to time the use of low fidelity effects, (iv) to detect user choice for passive haptics even when lacking physical props, (v) to sustain physical locomotion despite a lack of physical space, (vi) to reduce motion sickness during virtual locomotion, and (vii) to verify user understanding during story progression. We evaluated Mise-Unseen and our applications in a user study with 15 participants and find that while gaze data indeed supports obfuscating changes inside the field of view, a change is rendered unnoticeably by using gaze in combination with common masking techniques.

**Author Keywords**
Eye-tracking; virtual reality; change blindness; inattentional blindness, staging.

**INTRODUCTION**

*Mise-en-scène*, creating and arranging a scenery, or *staging*, is an essential part of many media productions, be it theater, games, or movies. Staging happens either before or, more interestingly, as an experience unfolds. While movies and plays follow a linear flow and predetermined timing of events, an unexpected need for staging may arise as a result of user input to an interactive system. Changes to the scene during staging are commonly designed to be hidden from users, requiring them to occur outside the visible area: They either happen behind an obstacle (e.g., door [13]) or simply

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.
outside the field of view [47]. Theater or movie productions make use of a controlled field of view to drive the narration – the user does not see the rigger or the boom operator, which would break the perceived consistency of the experience.

In virtual reality (VR) and augmented reality, dynamic staging poses a special challenge. First, the user may freely look around the scene and therefore might observe a change while it is happening, adding constraints to the application’s design. Second, as the field of view of headsets becomes larger, even less space will be available for dynamic staging. Today’s commercial VR headsets have a field of view of up to 120° horizontal and we expect them to eventually reach the maximum of human vision (~180° horizontal). Contrast this to traditional cinematography, where the director fully controls the viewpoint and camera focal length.

To create opportunities for dynamic staging in VR, we propose changing the content inside the users’ field of view. Since the user might notice such a change, breaking the user’s perceived consistency of the scenery, we use eye-tracking in combination with a variety of masking techniques (e.g., change saliency, distractors, etc.) to hide a change in plain sight. We believe that as eye-tracking becomes readily available in consumer VR devices (e.g., [19]) it can be used for dynamic staging in unattended areas (central vision is about 30° [52], foveal vision is about 5°).

We present Mise-Unseen, a software system to unnoticeably apply changes inside a VR user’s field of view to provide dynamic staging in VR. Our goal is to allow creators to author covert changes for dynamic staging. Mise-Unseen provides authored changes to any VR application. As illustrated in Figure 1 we show seven of such applications, pointing to possible use of this idea. In our user study, we validate our approach and show that hiding changes benefit from the use of eye-tracking together with masking techniques.

**RELATED WORK**

Mise-Unseen builds on work in eye-tracking, covert scene changes, perception theory (specifically change blindness and inattentional blindness), and stage magic techniques.

**Eye-tracking and covert scene changes**

Gaze information so far has been applied to use cases such as foveated rendering to save computational cost, more intuitive user controls [15,38], adaptive AI or new game mechanics [51] where eye-tracking is used for navigation or selection, displaying techniques [4], and for predicting users’ actions [12,45].

For our use case of effecting unnoticeable changes inside a user’s field of view, the related work is limited. Autopager [55] uses gaze to automate page turning in electronic documents or books by fading in new text outside the user’s fovea. Other studies show that covertly reducing image quality can be beneficial for foveated rendering [10]. Sparse Haptic Proxy [12] uses gaze to predict where users touch an object and updates its redirection methods on the fly to make repeated use of a prop for passive haptics. Sun et al. use saccades to shift user orientation during redirection [48].

**Covert scene changes and cognitive illusions without eye-tracking**

A range of applications achieve similar effects with head gaze instead of eye gaze. Redirection methods such as Impossible Spaces [47] as well as VMotion [46] change scenes to extend walking space. Sightline [60] changes scenes for a surreal cinematic experience. As these projects do not use eye-tracking, they update their scenes outside the field of view, whenever the user turns the head to look around.

Many applications apart from redirection methods implement cognitive illusions. HCI has long adopted cognitive illusions using stage magic techniques [20,49]. For example, Anderson et al. hide user interactions in collaborative tasks [1]. Marshall et al. deceive their users by adapting the ‘three cups’ magic trick [32]. Many other applications, including games, make use of such techniques [61].

Stage magicians use techniques that (mis)guide an audience’s attention, sometimes at the sensory level (e.g., smoke and mirrors). Cognitive illusions instead rely on social cues [29] and guiding user expectations [28] as Kuhn and colleagues explain based on their experience as stage magicians. Martinez-Conde and Macknik add that “cognitive illusions are not sensory in nature. Rather they involve high-level functions such as attention, memory and causal inference.” [33]. These high-level functions can be approximated with gaze data, as we will demonstrate in the system section.

**Perception theory and eye-tracking**

The aforementioned work on cognitive illusions and stage magic borrows from concepts in perception theory. Here we consider the potential impact of eye-tracking on models of visual attention, memory, and cognitive load (see [36] for an introduction).

**Visual attention measured by gaze**

Visual attention has been thoroughly studied (see [9] for a review). Arguably, the most important function of our selective visual attention is to direct our gaze rapidly towards objects of interest [24]. This evolutionary driven feature helps us perceive changes in the environment in a fast manner. While this ability would seem to make it very difficult to hide changes within the user’s field of view, research has demonstrated its limitations. Simons et al. found inattentional blindness [43] to occur during moments of attentional capture. Participants occupied with a primary task (counting ball throws) regularly failed to see an otherwise obvious stimulus (a gorilla passing through the scene). While a primary task seems to be essential to achieve inattentional blindness, some tasks do not direct gaze. Consequently, eye-tracking is beneficial in tasks that do not naturally direct the viewer’s gaze. One of our attentional models therefore accounts for the user’s gaze position.

**Visual memory measured by gaze patterns**

A viewer might fail to notice a change at the moment it happens, but recall previous staging and realize something has changed. Covert changes therefore require an understanding of user’s visual memory in order to exploit it. Change blindness is an effect similar to inattentional blindness where
viewers fail to notice a salient change in a scene. This effect is related to visual memory and human confidence. It occurs as a result of the richness of our daily visual experience [39]. To avoid semantic violations the brain prefers to simply assume the previously unregistered changes were part of the mental model from the beginning. In a way our outside world functions as our external memory [35], which can be overridden. Computational models for spatial memory exist [31] some of which include gaze information [39]. Ocular scan-paths are important for later recall [6,58], as well as path length and dwell time [21]. Considering the above, we implement a model of spatial memory to prevent later recall.

**Cognitive load measured by pupillometry and saccades**
Task involvement is an essential part of inattentional blindness. The pupil diameter, also measured by eye-trackers, can be used in this context. Pupil dilation has been used as an indicator of cognitive load [54] and several metrics exist that use pupillary activity to assess cognitive load [16]. However, there is controversy on its relation to inattentional blindness [57]. We implemented the Index of Pupillary Activity [16] to explore its potential use for detecting change, i.e., moments of higher probability of inattentional blindness. We further used eye-tracking to detect saccades, which can indicate user attention. User attention correlates with visual saliency [17], a spatial metric [24] for images and an excellent predictor for attention [7]. The distribution of visual saliency (or visual entropy) impacts the amount of saccades [50]. Few saccades can be interpreted as high workload or focus and therefore low probability of detecting a change.

**Covert attention and saccades**
It is possible to attend to something without directly looking at it. Such “covert” attention [41] would seem beyond the reach of eye-tracking based models. One of our models exploits the fact that intentional (endogenous) covert attention has a deploy time of about 300ms for specific locations and 500ms for specific features. Unintentional (exogenous) covert attention is transient and peaks at around 120ms after the event. Ideally, changes happen within those time limits, i.e., until 300ms after a saccade. More in-depth models of covert attention can be derived from gaze data [41].

**Supplementary masking techniques**
Changes made outside the fovea might still be detectable in many circumstances as we are drawn to visual motion and saliency [17]. This perceptual phenomenon has been used to direct gaze and attention [3]. To hinder attention, VR today incorporates different visual masking techniques [20]. The attentional blink [9,41], the inability to perceive distinct stimuli in close order, hinders temporal attention. A gradual fade of the stimulus hinders spatial attention [44]. We use these techniques to mask scene changes.

**MISE UNSEEN**
Mise-Unseen is a software system that applies covert changes to a virtual scene that can occur inside the user’s field of view. Mise-Unseen processes gaze data to prevent the user from noticing a change. Specifically it prevents observation of the change when it is happening, recall after it has happened and anticipation when it is about to happen (Figure 2). To achieve this, Mise-Unseen uses five models of user attention together with visual masking techniques. Each model incorporates the knowledge of a specific field of perception theory summarized in the related word. We will later discuss other forms of input besides gaze data.

![Figure 2: Mise-Unseen prevents anticipation, observation, and recall of a scene change using five attention models together with visual masking techniques.](image)

We will exemplify Mise-Unseen’s process with different applications. The jigsaw application (Figure 3) secretly rearranges its pieces to make solving it easier for the user. The forge application (Figure 4) places either a virtual hammer or an axe at the position of a physical surrogate for the user to experience passive haptics. In both cases the change should be hidden, as noticing unrealistic movement of weapons or jigsaw pieces would break immersion into the experience.

After the application defines what changes to inject into the scene Mise-unseen decides if and when to inject them. Mise-Unseen computes the probability of the user noticing it based on the application’s selected attention models (Figure 2). For example, the forge specifies that the hammer moves onto the physical prop using attention models one and five (current attention and spatial memory). Mise-Unseen computes each attention model’s value at every frame and takes the maximum as the probability of noticing the change. Once this value falls below a predefined threshold, Mise-Unseen injects the change into the scene and the hammer moves.

**Prevent observation: four models and visual masking**
Model one (current attention) implements the naïve approach: virtual objects that appear outside the fovea are less likely to be detected. Visual distractors, or masking techniques, reduce this likelihood further. The need for distraction may be eliminated by finding the moment where users are already involved in a task. Models two and three thus measure cognitive load, a requirement for inattentional blindness. While our applications did not use these more experimental models, we evaluated them in our study. Model four (covert attention) covers the possibility of attending to a changing object without looking at it.

**Attention model 1: gaze point for current attention**
We first compute the convex hull of a projection of an object’s geometry onto the view plane. If the angular distance
between the convex hull and the currently gaze point falls below a threshold, the object is considered to be in the user’s focus. This threshold depends on the precision and reliability of the eye-tracking data: we use a minimum angular threshold of 7° to be above the 5° angle of foveal vision [52], factoring in possible tracking errors.

Attention model 2: pupillometry for cognitive load
We implemented the Index of Pupillary Activity [16] as a measure of cognitive load. The advantage of using pupil dilation over absolute pupil size is that no calibration is required to establish a baseline. This measure uses fast wavelet transforms on the pupil diameter, counts local peaks of that value, and finally normalizes that count. We additionally factored in the scene illumination (various methods on the influence of luminance on pupil size exist [53]). Our model runs the same measure (fast wavelets) over the changing value for scene illumination. If pupillary activity occurs shortly after a spike in scene illumination (here 200ms), which could have triggered it, its signal is removed from the final count. While we did not find this signal alone to be a reliable predictor, its correlation with the detection rate of changes is evaluated in the user study.

Attention model 3: saccades for cognitive load
This model uses saccades as a measure of cognitive load (based on [17,50]). We count the number of saccades over time and normalize this number (using maximum of 5 saccades per 5sec).

Attention model 4: considering covert attention
Covert attention cannot be measured. This model merely reports its possibility based on measured saccades. From the moment of a saccade until 300ms after it was measured covert attention is impossible and the user focuses on objects inside the fovea only [41]. Applications may ignore this, trading a possibility for detection for 300ms less time delay. The jigsaw, for example, applies a due change also after a saccade instead of using a cross-fade. This is hard to combine with model three (saccades for cognitive load) – this model only allows for changes after a saccade, while model three recommends changes when fewer saccades occur, giving little opportunity for injecting a change. Covert attention can be prevented preemptively by implementing a primary task that requires the user’s visual attention (e.g., reading).

Visual masking techniques
Based on our literature review, we use the following techniques to mask transient signals:

(i) We present changes in unattended areas of the display, i.e., outside the user’s fovea.
(ii) We create salient noise that can create distractions to ‘drown’ the signal of the change.
(iii) We reduce the signal strength with visual techniques like gradual fades and low contrast.
(iv) We strategically provide tasks that will overload the attention of participants, preventing covert attention.

Applications may apply individual distractors by receiving a callback through Mise-Unseen’s API with a direction vector (pointing to scene coordinates onto which to distract the user) and the recommended size of the distraction (screen space). Our jigsaw puzzle responds to this callback by triggering an animation of the black cat on the table. Mise-Unseen also offers to inject a change when it would occur outside the field of view, behind another object, or during a blink or a saccade. To handle accidental gaze shifts Mise-Unseen either aborts or speeds up changes if they threaten to enter the fovea, depending on the duration left.

The masking techniques are an alternative to attention models two or three (Figure 2). The attention models are application agnostic, they make use of the fact that applications provide distracting tasks anyway. Masking techniques can be more reliable but the experience must be changed to effect them. They may constrain when or where the change happens, require certain visuals, or imply a user task. The moving cat we used fits into the theme of the jigsaw application, but cannot appear just anywhere in the user’s field of view.

Figure 3: Eye gaze driven adaptive difficulty. (a) To help the user solve this jigsaw puzzle by unnoticeably rearranging pieces, (b) such dynamic staging would normally be performed only out of view. Since users alone control the field of view in VR, Mise-Unseen detects the user’s gaze to (c) hide this change inside the field of view, but outside the foveated area. (d) As Mise-Unseen detects the user’s gaze resting on the cat, it cross-fades this puzzle piece to a new position (e) closer to its matching pieces. (f) The user continues solving the jigsaw puzzle, unaware that a change took place.
While a strong attention-grabbing event might work without gaze information, Mise-Unseen verifies that the user does not pay attention to a change, so that lighter, more casual motions can be used that better fit the design. These motions are easier to integrate or might already exist in the scene (moving shadows, wind effects). The natural movement of the cat is merely triggered by the system.

**Prevent recall: one model and staging recommendation**

Model five aims to prevent users from using visual recall to notice a change post-hoc.

**Attention model 5: dwell times for spatial memory**

We implemented a weighted directed graph to represent the user’s spatial memory, i.e., the distances (spatial and angular) between relevant objects in the scene the user has internalized. Figure 4d shows the forge as an example. Nodes represent all scene objects, e.g., weapons in the forge, as well as the user herself. The edges and their weights represent the internalized distance between pairs of objects (allocentric), and objects and user (egocentric). The egocentric weights increase when the user looks at an object, the allocentric weights increase when the user looks at any two objects in a row. The weights decrease when the user looks away (here after 1 second). As computational cost is exponential in the number of nodes, pairing all objects can be computationally costly. We reduce cost by working with a subset of relevant objects.

Mise-Unseen reduces the probability of a user noticing a change by recommending new positions and rotations for objects unconnected to the change. For example, the weapon in the forge that is not picked (Figure 4) moves together with the weapon that should be matched onto the physical prop. Mise-Unseen computes the difference of all internalized distances (edge weights) before and after the change, normalized by a baseline distance and baseline rotation. This difference quantifies the violation of spatial memory and is our predictor for attention. Mise-Unseen recommends a new position and rotation for each object. This results in less of a difference, less violation of spatial memory and thus lower probability of detection.
**Prevent anticipation: unpredictable triggers**

A user should not be able to anticipate a change by tying it to a specific, overt action: covert changes need covert triggers. We either use random triggers, or, more interestingly, derive verification of user understanding and user intention through eye-tracking.

The *jigsaw* uses random triggers, as it cross-fades pieces at random time intervals.

The *forge* uses user intent as a trigger. Once the user’s intent for reaching for a virtual hammer or an axe is clear. Mise-Unseen injects the change before the user takes action. We measured user intent by capturing and thresholding dwell times on objects (or sets of objects). This follows existing implementations that use eye-movement patterns to predict user actions [45,59].

The *hacker riddle* uses user understanding as a trigger. Once the user has understood the hint the riddle may be solved, else the hint changes. We measure user understanding by following related work [6,58] and match the user’s exhibited gaze pattern onto a pre-defined scan path (linear or tree). Here the user’s gaze must follow the hidden code sequence on a sheet of paper (Figure 5d). Our implementation uses an variation of attention model five (*spatial memory*) with faster increasing and decreasing edge weights, similar to a short term memory. Something is deemed understood once the edge weights closely match a set of pre-defined values.

**APPLICATION SCENARIOS**

The main motivation of this work is showing the feasibility of hiding changes in plain sight, yet outside the foveated area. We implemented seven applications, depicted throughout the paper. Each application shows a distinct use-case of our approach and the feasibility of hiding changes in this context. Applications are similar in so far as they share the same attention model(s). For example, the *forge* and the *gallery* both use attention model five, but the former enables passive haptics, while the latter individualizes content to the user.

Two applications (*forge, loading room*) are specific to VR, while the others apply to desktop environments as well.

**The Forge – free choice in passive haptics**

Figure 4 shows and describes this application. Related work on passive haptics [23] includes retargeting methods for prop re-use [2, 12] and dynamic adaptation of virtual objects onto props [22]. This application shows that instead of retargeting user movements [12], the idea of adaadaptation can use eye-tracking as an indicator for user’s.

**The Loading Room – free choice in real-walking**

Figure 6 shows and describes this application. The *loading room* is conceptually similar to the *forge*. Instead of matching weapons onto a prop, one of two walking paths are matched onto space. Space compression techniques already include the adaptation of the virtual world onto the physical world [34,42], even by using change blindness outside the user’s field of view [47]. Also, the idea of predicting walking directions is not new [59]. This application illustrates how these approaches could benefit from eye-tracking to create more opportunity for these changes. As changes for real-walking would use more screen space than changes for e.g. passive haptics, distractors are mandatory, possibly using other modalities such as audio (e.g., phone ringing).

![Figure 6: Real-walking enabled by eye-tracking](image)

**The City Flythrough – cinematic sequences, 360 video, virtual locomotion**

Figure 7 shows and describes this application (video contained in auxiliary material). In contrast to the black masks that are commonly used to eliminate motion in the peripheral vision, we maintain a full field of view. We instead remove peripheral motion by reducing the frame rate outside the fovea to 1Hz and blending the renderings of foveated and peripheral areas together. We hide the reduced frame rate in the peripheral area by interpolating between frames, feathering the edge to the foveated area, and adding a motion blur.

![Figure 7: Virtual locomotion](image)
Ultimately the effect is similar to foveated rendering, but instead lowers the time resolution instead of pixel resolution and uses frame interpolation instead of pixel interpolation. Because unlike the other proposed techniques this change is not transient in character, we evaluated it separately (see ‘reported noticeability’ in results).

The Jigsaw Puzzle – adaptive difficulty
Figure 1 and, in more detail, Figure 3 show and describe this demonstration of adaptive difficulty. For cognitive tasks, designers might want to provide help in a covert way so as to not diminish the user’s satisfaction in completing a task.

The Gallery – adaptive content
Figure 8 shows and describes a demonstration. Past work has demonstrated how to change a presentation or personalize an experience using gaze [11,62]. The gallery builds on this idea and hides the changes. This illustrates how adaptive content benefits from our approach as obvious and unbelievable transitions can reduce immersion.

The Hacker Riddle – validating user choices / story progression
Figure 5 shows and describes this application. It shows how applications can covertly prevent users from guessing their way through an experience. This may help foster story progression or support a narrative arc, e.g., in adventure games, or it may be used in training or learning applications.

The Car Explosion – saving development costs
Figure 9 shows and describes this application. Foveated rendering already saves computing cost. We want to highlight that also development costs can be reduced, leaving more resources for development of other parts of the experience. As users tend to afterwards look at the area where the transient change happened, we guide user’s focus. This guiding of users attention points to possible use for authoring, e.g., to achieve “synchrony of attention” in narratives [5].

Figure 9: Saving development costs. (a) The user does not attend to this car. (b) Instead the user attends to the street sign the moment this “explosion” happens. The low fidelity of this effect is harder to perceive as it occurs outside the user’s fovea. (c) The user looks at the car after it exploded.

IMPLEMENTATION
The software, example applications and user study are implemented in C# using Unity3D. Relevant source code is available online for researchers to replicate our work [63].

USER STUDY
We validated Mise-Unseen in a two-part user study.

We first conducted a psychophysics study to better understand how gaze data supports hiding changes in the participant’s field of view. We implemented a visual search task [56], in which participants identified a new target dot appearing within a cluster of 8 other dots (Figure 10). We compared a gaze-only method to a baseline condition, in which the dot appeared in the center of the field of view, and five other off-fovea experimental conditions that use different visualization techniques to make the visual search task more difficult.

The second part of our study demonstrates the external validity of these results. Since our applications use similar visualization techniques, direct comparisons between applications should follow results of the first part.

Participants
We recruited 15 participants from our organization (5 female, 10 male, mean age 36.8 sd 10.1 years). Of these participants, 12 had experience with VR, 5 with eye-tracking and 8 with magic tricks. All participants gave written informed consent (according to the declaration of Helsinki) and were paid for their participation. This user study was approved by an Institutional Review Board.

Interface conditions
In the first part of our study we compared a total of six conditions, each motivated by pilot studies which we do not report.

In our baseline condition, the target appeared in the center of the field of view (head-gaze), depicting the worst case scenario of placing the change. To not appear on the center of the display we added a random $5^\circ$ angular offset.

In our off-fovea condition, the target appeared with a $33^\circ$ angular offset to the head gaze just inside the field of view, but outside the fovea. The target was randomly placed within a $180^\circ$ radial range opposite to the currently measured eye gaze, thus making the position of the target unpredictable to the user (see Figure 10). Effectively, this puts the target out
of the foveal vision of $5^\circ$ and central vision of $30^\circ$. Compared to the baseline, this condition shows the effect of gaze on hiding scene changes.

Covertness of scene changes is increased using visual masking techniques represented by the following conditions:

In our add-crossfade condition, the target appeared similar to the off-fovea condition, but with a linear three second fade-in.

In our add-low-contrast condition, the target appeared similar to the add-crossfade condition, but the final color was altered to use 80% of the background color, lowering the contrast of the stimulus to the background.

In our add-task condition, the target appeared similar to the add-low-contrast condition, but the participant was faced with an RSVP (rapid serial visual presentation) reading task [25]. The participant was forced to read the text out loud as a primary task while searching for the target (text appeared in front of the participant with ~14 characters/sec using IELTS text samples “lessons from the titanic”).

In our add-motion-distractor condition, the target appeared similar to the add-low-contrast condition, but the set of other dots was increased from 8 to 18 and moved around the display with varying speeds.

In the second part of our study, we compared five of our applications which exemplify the previously experienced visual search tasks.

The forge and gallery experiences demonstrate the add-crossfade condition, as objects fade in and out, but neglect differences in contrast and provide no primary task.

The hacker riddle experience demonstrate the add-task condition, as participants were tasked with finding the missing code sequence, but no motion distractor was used.

The jigsaw experience demonstrates the add-motion-distractor condition, as a primary task was given (solving the jigsaw), and the cat provided added motion whenever the pieces moved. Additionally, the contrast of the pieces to the table was lowered.

We evaluated our virtual locomotion technique using two separate conditions in the city application.

In our city black mask condition (Figure 7b), the field of view was diminished using the classic circular cutout used in many VR applications, with the difference that the cutout followed the participant’s gaze. This is equivalent to the off-fovea condition, as there is no primary task, and only gaze is used for obfuscating the visualization.

In our city full-FOV condition (Figure 7c), the participant had a full field of view, but motion outside the fovea was removed by reducing the update rate to 1Hz. We cross-fade between frames and add motion blur to hide the reduced update rate. This is equivalent to the add-low-contrast condition, as contrast between areas inside and outside the fovea is lowered and cross-fades obfuscate the visualization.

---

**Apparatus**

We conducted the study in our lab, using a VIVE Pro HMD and Trackers to track participants’ head and hand as well as the prop, a PupilLabs system for eye-tracking (200Hz, 4.5ms latency) and the applications described earlier.

**Task and procedure**

After being introduced to and consenting to the experiment, each participant calibrated the eye tracking system. The first part of the study took roughly 15 minutes, the second part 30. Participants received compensation for their effort.

In the first part, for each trial participants had to find a target dot appearing in a cluster of other dots that were placed within a range of a horizontal 120° and 66° from the participant (Figure 10). Participants could move their head and eyes freely. Participants were given a brief tutorial of three trials. The position of the other dots changed each trial. Participants had to turn their head towards all dots before the stimulus was introduced, to control for the possibility of visual recall (except add-motion-distractor that had more dots).

The timing of the onset of the stimulus was randomized, as was the order of trials. Participants identified new dots by turning their head towards it and pressing a key. We chose a within-subject design with an adaptive number of repetitions per conditions (12 to 18, see [37], also see auxiliary material for a complete walkthrough).

![Figure 10](image)

Figure 10: (a) In our visual search task our study participants saw a cluster of dots in which they had to identify an appearing dot, like (b) this red dot. (c) The stimulus is randomly placed (plus sign denotes head gaze, minus sign denotes eye gaze not visible to participant).

For the second part of the study, participants experienced a random selection of the applications described in the user study section above (due to time constraints), during which they also gave qualitative feedback by thinking aloud.

**Measurements**

We measured detection rate in the first and enjoyment and reported noticeability in the second part of the study.

The detection rate, our main metric, is the ratio of correctly observed newly appearing dots. For small differences in detection rate we additionally compare inverse-efficiency (see [8]), the ratio between task completion time (time on identification of stimulus minus time on stimulus onset) and the detection rate. Low values indicate good task performance.

Enjoyment was measured by asking participants to rate a 7-item Likert scale statement (“I enjoyed this experience” 1 – agree not at all, 7 – agree very much). This metric was taken to show that the implemented applications are valid; we merely report these values.
Reported noticeability was measured similarly (“changes in the virtual scene were noticeable” 1 – agree not at all, 7 – agree very much). This metric was taken for comparisons.

Hypotheses
We compared the off-fovea condition to the other five conditions, including the baseline condition. We hypothesize that changes in the off-fovea condition are less noticeable (lower detection rate or higher inverse efficiency) than in the baseline condition (H1), but more noticeable than in the conditions add-crossfade (H2), add-low-contrast (H3), add-task (H4) and add-motion-distractor (H5). Additionally, we want to see if the psychophysics results are consistent with the applications part of the study. We assumed that noticeability in the jigsaw would be lower than hacker riddle due to motion distractors (H6), which in turn would perform better than gallery and forge due to the primary task (H7). The stimulus in the city is not transient, making it hard to compare to the others, so we merely hypothesized the city full-FOV is less noticeable than city black mask (H8).

Analysis
We used a Friedman test for the analysis of the visual search task with two-tailed Wilcoxon signed rank tests for post-hoc pairwise comparisons. For the second part, in which each participant experienced a different subset of applications, we ran a Kruskal-Wallis together with Wilcoxon rank sum tests.

Results
Detection rate: Figure 11 shows the main results. Detection rate was significantly different between conditions ($\chi^2(5) = 38.5$, p < .001). Detection rate is lowest in add-task and add-motion-distractor. Add-motion-distractor was significantly lower than add-task (p < .05, r = .64). When compared to off-fovea, both conditions show lower detection rate, (add-task: p < .05, r = .51; add-motion-distractor: p < .01, r = .84). This confirms H4 and H5.

Figure 11: Detection rate is significantly lower for conditions involving a task and moving distractors (bars show standard error).

The remaining conditions do not show significant differences on detection rates, despite the fact that detection lowers incrementally with the added complexity of the different conditions. For such smaller detection differences we compare their inverse efficiency.

Inverse efficiency: Figure 12 shows the results. Inverse efficiency was significantly different between conditions ($\chi^2(3) = 16.4$, p < .001). Since no successful detections occurred in the add-motion-distractor condition and only a few in the add-task condition, only the first four conditions are included for computing the main effect. We observe that off-fovea has a significant impact on inverse efficiency of the visual search task when compared to baseline (p < .05, r = .74), i.e., using gaze data reduces task performance for detecting changes. Add-crossfade is not measurably different, but add-low-contrast increases inverse efficiency significantly more than off-fovea (p < .05, r = .85). These results do not confirm H2, but confirm H3 and, importantly, H1.

Figure 12: Inverse efficiency of the off-fovea condition is higher than in the baseline. Adding visualization techniques obfuscate the injected change further.

Results from the second part of the study validated the applicability of the psychophysics results in our demonstration applications.

Reported noticeability: Figure 13 shows the results. Reported noticeability was significantly different between conditions (H(5) = 20.6, p < .001). Participants found the city black mask more noticeable than city full-FOV (p < .05, r = .61) confirming H8. Hacker riddle performed better than forge (p < .05, r = .51) and gallery (p < .05, r = .54) confirming H7. Jigsaw scored lower than the hacker riddle, however, not significantly, therefore we cannot confirm H6.

Figure 13: Participants average ratings on the statement “changes in the virtual scene were noticeable” reflect the findings from the first part of the study (bars show standard error).

Enjoyment: Overall, participants enjoyed our applications the jigsaw (5.2 sd 1.3), the hacker riddle (4.9 sd 1.0), the gallery (5.2 sd 0.7), the forge (6.5 sd 0.5), and the city (4.1 sd 1.7).
Qualitative feedback

Participants noticed some changes in the gallery condition (P4: “I noticed the gradual transition”, similar P9), often due to recall (P1: “the moment I looked back”, similar P4).

Participants reported to have noticed less changes in the jigsaw condition, because of the primary task (P14: “I was more concentrating on how to complete it”), similar P1, P2, P6, P8) and because of the distractor (P12: “the cat was moving, I was not looking at anything else”, similar P5).

Most participants preferred the city full-FOV condition (P5: “there is no motion sickness”, similar P5, P6, P7, P15), but some did not (P10: “the update rate is noticeable if objects are close”, similar P9, P11). Participants commented on the field of view (P12: “in [black mask] the field of vision decreased”, similar P8) and the contrast (P15: “[black mask] I am drawn to the contrast”, similar P4, P5, P6, P7, P11, P12).

Additional results – cognitive load measured by pupillometry and amount of saccades

Pupillometry has been proposed to indicate a user’s attention to a task [26] or cognitive load due to task involvement [16]. We describe our implementation of this measure of cognitive load in our system section (model two). We computed the average of this measure from the onset of the stimulus until the end of the trial. We found that cognitive load correlated positively with our inverse efficiency metric (p < .05, r = .38), i.e., negatively with task performance. Changes are less noticeable when cognitive load increases. The baseline condition was removed from this analysis as it showed outlier characteristics (data points too close together due to high efficiency). This correlation between the pupillometry and our task performance further validates the use of this metric for cognitive load. We also examined our saccade-based metric of cognitive load (model three), but found no conclusive results.

DISCUSSION

Our main insight is that gaze data supports hiding changes inside the user’s field of view. The use of attention models paired with masking techniques, such as occupation with a primary task (hacker riddle) or presentation of visual distractors (jigsaw), reduces the detection rate to a level that may be useful in many applications.

We think these results can be generalized. Consistent with our measurements, participants reported less noticeability in applications with these masking techniques. We believe our results are ecologically valid, as participants could move their head and eyes freely and reported high enjoyment.

Our study concentrates on Mise-Unseen’s prevent observation part. It directly validates attention model 1 (current attention) and our visual masking techniques. It also shows that attention models for deriving cognitive load could potentially be used to supplement or replace visual masking techniques. Specifically the pupillometry data (model 2), while noisy, shows noteworthy preliminary results. Using saccades (model 3) looked promising during pilots, but we could not measure a correlation in the study. We did not run a study on covert attention (model 4), but we found it to work better with changes small in screen size (e.g. hacker room), as cross-fades cannot be used due to time constraints. We propose Mise-Unseen’s other parts prevent anticipation and prevent recall with our computational model of spatial memory (model 5) as a stand-in for possibly much better computational models [31,39]. These should be evaluated separately.

We speculate that Mise Unseen will find use in a broad range of fields. VR will be ubiquitous [13,34,46,47], but tracking data of the physical environment will include noise and delay, so runtime changes for dynamic staging are required. Ubiquitous VR will, however, first require further research on parametric and generic design. We presented other uses like detecting user preference or logical understanding [40]. This may have implications for storytelling as advances in narrative are now possibly contingent on where one looks. It also extends Mise Unseen’s applicability to areas like education and training applications where task adherence is important. Since users do not notice they are being helped, motivation might increase. On a hardware level, the quality of eye-tracking will further improve and compensate for drift [18] (our eye-tracker has an accuracy of 1°, but a lack of auto calibration caused drift, requiring occasional recalibration). More importantly, the field of view in future VR headsets will increase and more space for dynamic staging will be available, supporting our approach.

In this work we focused on eye-tracking. Naturally, our approach can be fused with other input. Video processing provides saliency maps, which are great indicators of attention [7]. Brain-computer interface signals, like EEG, could also supplement our approach [27,30]. Our visual masking techniques can be extended (e.g., [3]). On a mechanical level, a change could be triggered while forcing a user to blink [14].

CONCLUSION

We presented Mise-Unseen, a software system for covertly injecting changes to virtual scenes inside the user’s field of view. We incorporated findings from our literature review to process gaze to form high-level models of attention, intention and spatial memory to find out if and when to inject a change into a scene. We use Mise-Unseen in our implementation of seven different applications that we presented in this paper to highlight potential usages for the system. We validated our system and our applications in a two-part user study. We found that eye gaze indeed supports hiding changes, but only in combination with different masking techniques can changes be obfuscated reliably.

REFERENCES


[2] Mahdi Azmandian, Mark Hancock, Hrvoje Benko,


[7] Raymond Bruyer and Marc Brysbaert. 2011. Combining speed and accuracy in cognitive psychology: is the inverse efficiency score (IES) a better dependent variable than the mean reaction time (RT) and the percentage of errors (PE)? Psychologica Belgica 51, 1: 5–13.


