

StateScribe: Towards Accessible Change Awareness Across Real-World Revisits

Ruei-Che Chang*
University of Michigan
Ann Arbor, MI, USA
rueiche@umich.edu

Xirui Jiang*
University of Michigan
Ann Arbor, MI, USA
xirui@umich.edu

Rosiana Natalie
University of Michigan
Ann Arbor, MI, USA
rosianan@umich.edu

Hao Chen
Samsung Research America
Plano, TX, USA
hao.chen1@samsung.com

Vlad Roznyatovskiy
Samsung Research America
Plano, TX, USA
vlad.r@samsung.com

Jianzhong Zhang
Samsung Research America
Plano, TX, USA
jianzhong.z@samsung.com

Kang G. Shin
University of Michigan
Ann Arbor, MI, USA
kgshin@umich.edu

Ke Sun
University of Michigan
Ann Arbor, MI, USA
kesuniot@umich.edu

Anhong Guo
University of Michigan
Ann Arbor, MI, USA
anhong@umich.edu

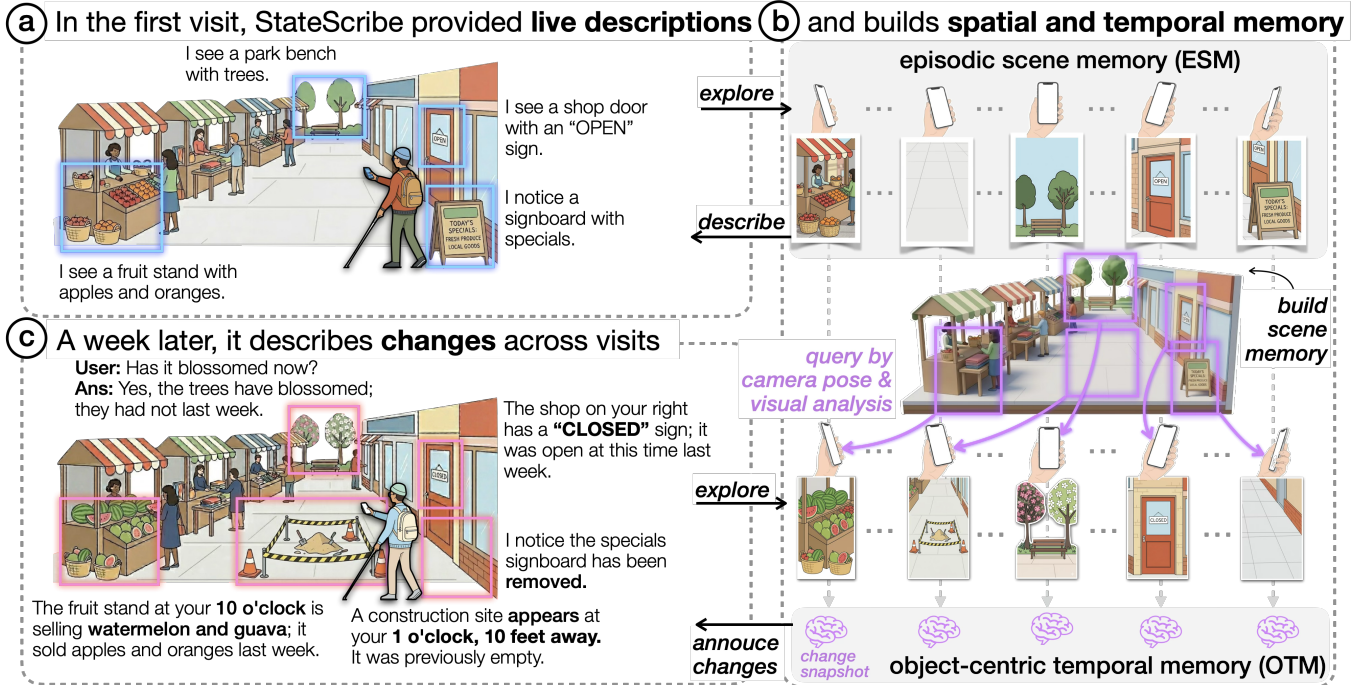


Figure 1: StateScribe enables accessible change awareness across real-world revisits. When visiting a new location, (a) StateScribe provides live descriptions during exploration to support scene understanding. Simultaneously, (b) it builds spatial and temporal memory through a novel dual-layer memory architecture, including an Episodic Scene Memory (ESM), which stores recent scene observations, and an Object-Centric Temporal Memory (OTM), which tracks object-level changes over time. Upon revisiting the same location, (c) StateScribe retrieves and compares past views via camera pose and visual similarity, and announces changes that have occurred since the previous visit, including object appearance, removal, replacement, or content modifications.

*Both authors contributed equally to this work.



This work is licensed under a Creative Commons Attribution 4.0 International License.
UIST '26, Detroit, MI, USA

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2856-3/2026/11
<https://doi.org/10.1145/3830398.3830475>

Abstract

Real-world environments evolve continuously, yet blind and low-vision (BLV) individuals often have limited access to understanding how they change over time. Unexpected or relocated objects, layout modifications, and content updates (e.g., price changes) can introduce safety risks and cognitive burden. While existing visual assistive technologies can describe immediate surroundings, they

operate as one-off interactions and lack mechanisms to surface meaningful changes across revisits. Informed by a survey of 33 BLV individuals, we develop StateScribe, a system that supports accessible awareness of real-world changes across revisits. StateScribe employs a dual-layer memory architecture that integrates episodic scene memory and object-centric temporal memory to enable scalable and structured change tracking. It provides both live descriptions of the current scene, and descriptions of *what* has changed, *when* and *where* it occurred across revisits, such as “*The shop on your right has a “CLOSED” sign; it was open at this time last week.*” Our evaluation shows that StateScribe maintains high accuracy (F_1 -score = **83.1%**) across 11 revisits, while remaining low-latency (mean<**1.42s**) and memory-efficient (<**55MB**) across 110 revisits. A user study with nine BLV participants demonstrates that StateScribe improves change awareness across revisits in three real-world locations. Finally, we discuss implications for long-term AI-assisted companions that support broader change observation using multimodal sensing, extend beyond changes to other memory capabilities, and adapt to individual users, intents, and contexts.

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**; **Accessibility systems and tools**.

Keywords

Visual descriptions, blind, low vision, visually impaired, assistive technology, accessibility, memory, RAG, LLM and VLM, agent

ACM Reference Format:

Ruei-Che Chang, Xirui Jiang, Rosiana Natalie, Hao Chen, Vlad Roznyatovskiy, Jianzhong Zhang, Kang G. Shin, Ke Sun, and Anhong Guo. 2026. StateScribe: Towards Accessible Change Awareness Across Real-World Revisits. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26)*, November 02–05, 2026, Detroit, MI, USA. ACM, New York, NY, USA, 22 pages. <https://doi.org/10.1145/3830398.3830475>

1 Introduction

Real-world environments evolve continuously with many unanticipated changes, yet blind and low-vision (BLV) individuals often have limited access to information that helps them understand how familiar spaces transform over time. For example, newly introduced obstacles, relocated objects in shared spaces, content changes (e.g., price updates), or layout modifications can pose safety risks and increase cognitive burden. Despite these challenges, there remains a limited understanding of what constitutes meaningful real-world changes for BLV individuals, and how to convey them effectively.

While existing human-powered services (e.g., Be My Eyes [3], Aira [2]), AI-powered visual assistive technologies (e.g., Seeing AI [6], Be My AI [1], WorldScribe [15]), or live video AI (e.g., ChatGPT Live Video [7], Gemini Live [4]) can describe immediate surroundings, they typically operate as one-off sessions that retain only within-session memory. As a result, they lack persistent 3D spatial memory of a location [17] and mechanisms for surfacing meaningful changes across revisits. Enabling such longitudinal assistance poses several challenges: (i) **Maintaining longitudinal memory in 3D space is difficult**, as systems must identify and

store the most relevant observations from the 3D real world without overloading with redundant information. (ii) **Scaling memory storage and retrieval over extended use is difficult**, as spatial and temporal data grow continuously, increasing retrieval latency and confusion. (iii) **Incorporating change-awareness into real-time interactions can easily overwhelm users**, especially if updates are frequent and noisy.

Based on a survey of 33 BLV individuals, we identified categories of changes that are difficult to perceive yet significantly affect daily life, such as newly introduced obstacles in shared spaces (e.g., scattered packages), changes to signage along routine commuting routes (e.g., removed or added signage), and updated pricing or layout during grocery shopping. To address these, we present StateScribe, a system that supports accessible awareness of real-world changes through a smartphone and its sensor data. StateScribe employs a dual-layer memory architecture that integrates Episodic Scene Memory (ESM) and Object-centric Temporal Memory (OTM). Specifically, ESM continuously updates information from recent visits within a context window, archiving older data to maintain an up-to-date representation of the scene. In contrast, OTM selectively maintains key objects, tracking their states and spatial locations across visits while recording meaningful changes. This design enables scalable, structured, and efficient storage and retrieval.

Using these memory structures and smartphone 3D data (e.g., RGB-D images and camera intrinsics), StateScribe retrieves ESM and OTM by querying the current camera pose to identify the most recent frames from prior visits and detect significant changes. It generates descriptions that clearly communicate *what has changed, and when and where the change occurred*. For example, a user revisiting the market (Figure 1b) a week later might hear: “*The fruit stand at your 10 o’clock is selling watermelon and guava; it sold apples and oranges last week,*” “*The bakery on your right now shows a ‘CLOSED’ sign; it was open at this time last week,*” or “*A construction site is at your 1 o’clock, 10 feet away.*” Beyond change descriptions, StateScribe also provides live descriptions (Figure 1a) to support real-time scene understanding, using a similar but simplified approach to WorldScribe [15]. StateScribe further enables interactive exploration of how their environment has evolved, where users can query via speech, e.g., “*Has it blossomed now?*”, with responses: “*Yes, the trees have blossomed; they had not last week*”(Figure 1b).

In a technical evaluation against two baselines, a *Live model* that actively prompts for change detection and a *Offline video model* that processes full-visit videos, StateScribe achieved an F_1 -score of **83.1%** across **11** visits (291 changes), outperforming the *Live model* (40.1%) and *Offline video model* (27.4%). It also showed higher spatial accuracy, with lower clock-direction errors (in hour units; **M=0.24** vs. 1.98 and 1.86) and smaller distance estimation errors (in feet; **M=0.68** vs. 5.29 and 6.49). Additionally, StateScribe remained low-latency (**mean<1.42s**) and memory-efficient (<**55MB**) across **110** revisits in simulated long-term use. A user study with 9 BLV participants showed that StateScribe enhanced change awareness across revisits in different settings (e.g., office, grocery store, courtyard).

Finally, we discuss extending StateScribe to video-based assistive technologies, including remote sighted assistance (RSA) services [2, 3] and live video AI systems [4, 7], and derive design implications for multimodal, personalized, long-term AI companions with broader memory capabilities that adapt to users, intents, and contexts.

2 Related Work

StateScribe builds on prior work to provide BLV people with visual descriptions for real-world understanding, as well as traditional and AI-enabled memory-augmented systems. In what follows, we describe our motivation from prior related work and research gaps.

2.1 Visual Descriptions for BLV People

Textual descriptions are essential for making digital and physical information accessible to BLV people. Recent advances in VLMs enabled automated image descriptions and conversational interaction, reducing reliance on human-authored alt text. These capabilities allowed BLV users to quickly obtain visual summaries and interactively explore details, which are increasingly integrated into social network platforms (e.g., Facebook [72]) and assistive apps (e.g., Seeing AI [6], Be My AI [1]). In addition, prior work has explored integrating VLMs into systems that provided continuous, live visual descriptions. Such systems dynamically adapt the level and content of information based on user behavior, including camera motion [15] and hand movements [18].

Beyond visual descriptions, prior work in computer vision developed models for detecting and summarizing visual changes across images [26, 39]. In parallel, HCI researchers explored systems that describe visual differences between image pairs to support an accessible collaborative slide editing [60], image editing [16, 36, 75], and graphics and 3D geometry editing [76, 77]. However, support for detecting and describing real-world visual changes remains limited. Environments change continuously and unpredictably, which can impact BLV individuals' daily lives. For instance, unexpected barriers in outdoor environments [21, 22, 71], relocated objects in indoor shared spaces [59, 69], or seasonal product rearrangements and price updates in retail environments [41, 45, 68, 74] could potentially introduce safety risks and cognitive burdens for BLV individuals. To close these gaps, we first categorized meaningful changes through a formative study and developed StateScribe, a system that possesses spatial and temporal memory to detect and describe meaningful changes across visits on familiar routes or locations, beyond live descriptions of immediate surroundings.

2.2 Human Memory-Augmented Interactions

Early work in human memory-augmented interactions [42] took the form of lifelogging systems, enabling applications such as social or audio reminders [30], contextual suggestions [62], sharing everyday moments [30], and support for people with memory impairments [32, 46, 47]. While effective, these systems faced key challenges related to capturing meaningful data, managing storage constraints, and enabling timely retrieval of past data.

First, they typically relied on continuously capturing large portions of everyday experiences across modalities, including photos [28, 29, 32, 55], videos [33, 54, 63], audio signals [30, 62, 70, 80], GPS data [28, 29, 33, 62], and physiological signals [11–13, 31, 33, 40]. However, such continuous recording imposed substantial storage demands, limiting long-term usability. To mitigate this, some systems adopted selective capture strategies, such as manual recording [42], biosignal-based triggers (e.g., startle responses [31] or emotional changes [40]), and spatial triggers [29] (e.g., remaining within a 50-meter radius for over five minutes). Beyond storage, accessing

and making use of captured data remained challenging. Traditional interfaces primarily supported browsing by time [20], location [67], or both [19, 28, 29], offering limited support for efficient browsing and retrieval of relevant past information.

Recent advances in multimodal large language models (MLLMs) and retrieval-augmented generation (RAG) have improved how memory is captured, stored, and retrieved. Prior computer vision work explored memory structures for embodied agents across tasks such as spatial reasoning [79], robot navigation [34], and persistent 3D scene understanding [25], but these were optimized for offline accuracy rather than real-time human-AI interaction. In contrast, systems like Memoro [80] store conversational snippets as lightweight memory units for efficient live retrieval, while OmniQuery [50] extracts semantic signals from images and videos for structured indexing. Memory Reviver further organizes photo collections into hierarchical representations to support conversational retrieval for BLV users [73]. However, these largely text-based approaches remain limited in dynamic real-world settings, where rich visual information cannot be fully captured through text alone.

To address this, StateScribe employs a novel memory architecture that captures visual information from camera frames and selectively retains meaningful object changes in both visual and textual modalities across recent visits, while archiving outdated data to maintain lightweight storage for extended use. StateScribe also has an efficient memory storage and retrieval pipeline that leverages camera poses and visual analysis, enabling it to proactively announce changes in real-time and across visits.

3 Formative Study: Understanding Real-World Changes Meaningful to BLV People

To address the key question, *What constitutes meaningful real-world changes for BLV people?* We surveyed 33 participants (16 blind, 17 low vision; 10 men, 21 women, 2 non-binary; mean age = 45.1, SD = 15.9) recruited through local organizations and prior contacts (Table 1). We analyzed responses using affinity diagramming.

3.1 Results

3.1.1 Changes during daily commutes. Participants frequently reported **temporary barriers** near construction sites (N=28), such as fences, traffic cones, scaffolding, tape, and detours, which obstructed their usual routes. The construction sites also sometimes **removed or obscured key landmarks** (N=7), including signage, pedestrian signals, and tactile paving. Also, natural changes, such as snowbanks, flooding (N=23), and fallen or overhanging tree branches, further disrupted familiar routes. Furthermore, the **replacement of familiar landmarks** caused confusion, as F15 noted: “A bus stop that used to be just a bench was replaced with a metal shelter. I didn’t expect it and bumped into it.” While some high-level changes can be anticipated (e.g., via weather or transit updates), pedestrian-level changes remained difficult to detect, with little warning (F20: “no commotion or construction noise”). Participants often noticed them only after collisions, cane detection, or others’ alerts.

3.1.2 Changes during grocery shopping. Beyond **temporary barriers** (e.g., carts or displays blocking aisles), participants reported **spatial changes** during grocery shopping, including product relocations (N=24) and aisle rearrangements (N=21). These unexpected

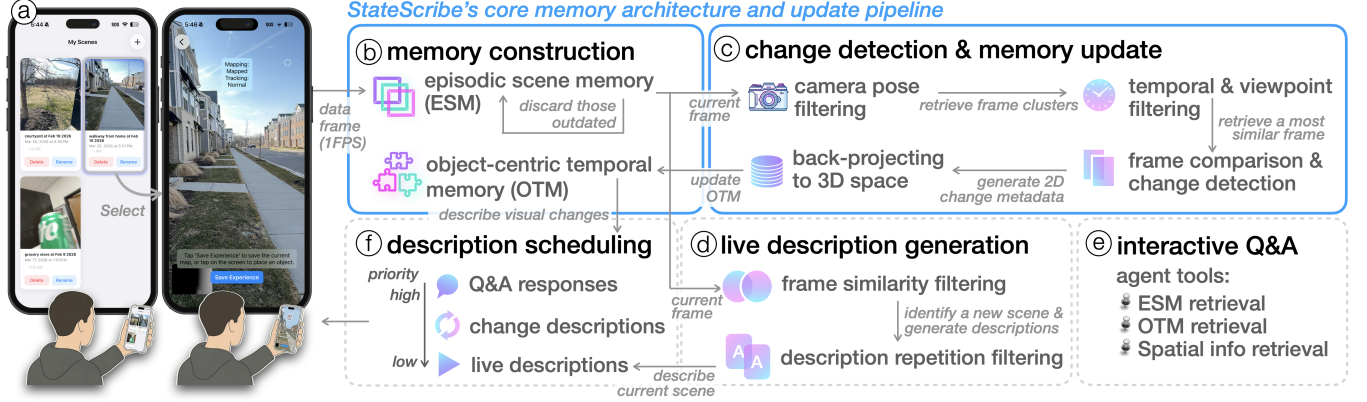


Figure 2: StateScribe's system architecture and processing pipeline. (a) StateScribe's mobile interfaces. (b) StateScribe constructs episodic scene memory (ESM) from streaming frames and updates object-centric temporal memory (OTM) through a change-detection pipeline. (c) For each incoming data frame (consisting of RGB-D images, camera intrinsics), StateScribe retrieves the most relevant reference frame and compares it to generate change metadata, which is then used to update the OTM. (d) In parallel, ESM supports live scene description by filtering visually similar scenes and avoiding repetitive descriptions. (e) Users can ask questions about the current scene, past scenes, or detected changes; a Q&A agent answers these queries using built-in tools such as object localization and retrieval of recent scenes or changes. (f) Descriptions are scheduled and delivered based on priority: Q&A responses, change descriptions, and then live scene descriptions.

changes caused frustration (F18: “The chip aisle became canned food that I had to find someone to locate it.”). Also, **price changes and sales signage** (N=19) often went unnoticed, leading to surprise at checkout (F6: “I defaulted to the old price.”). To cope with these, some used RSA services (e.g., Aira [2]) or AI-powered apps (e.g., Seeing AI [6]), occasionally scanning the environment to locate items (F28). However, most relied on nearby shoppers or store staff and expressed a desire for advance awareness of such changes.

3.1.3 Changes within the home. Participants frequently encountered **temporary obstacles or clutter** (N=26), including blocked pathways (N=16), furniture rearrangements (N=11), and seasonal decorations (N=9). Such changes stemmed from both participants (N=9) and other household members, with most living with family (N=14) or roommates (N=7). For example, F20 noted that everyday household behaviors (e.g., leaving boxes on the floor) introduced disruptive clutter. These disruptions were particularly challenging as many participants did not use a cane in familiar home environments, leading to collisions (F28: “A chair left in my walking path caused me to fall.”). While a few low-vision participants could detect changes under good lighting, most became aware only through physical contact, often resulting in frustration or minor injuries.

3.2 Summary of Meaningful Visual Changes

In sum, participants mainly relied on memory or direct encounters to navigate familiar environments, occasionally using assistive tools (e.g., white canes, mobile AT) to manage unexpected changes. However, awareness remained largely reactive, highlighting the need for proactive information and advance alerts to reduce mobility difficulties, hazards, and frustration. Based on these insights, we categorize meaningful changes as follows:

- (i) **Object-Level Changes:** Changes involving the presence or the identity of individual objects at a location.
 - **Appeared:** Newly introduced objects not previously present (e.g., construction barriers, clutter).

- **Removed:** Previously existing objects that are no longer present (e.g., missing landmarks or signage).
- **Replaced:** Objects substituted by different ones at the same location (e.g., a bench replaced by a bus shelter).
- (ii) **Spatial Changes:** Changes that alter the arrangement or navigability of the environment.
 - **Relocated:** Objects moved to new locations (e.g., product relocations in stores).
 - **Layout Changes:** Modifications to spatial layout that affect mobility (e.g., aisle rearrangements, blocked pathways).
- (iii) **Attribute Changes:** Changes in the properties or state of existing objects without altering their identity or position (e.g., price changes, signage updates, on/off status).

4 StateScribe

StateScribe¹ is a system with spatial and temporal memory that observes and describes real-world changes across visits, going beyond live descriptions of the current scene. For example, when a BLV user explores a new market (Figure 1), StateScribe provides live descriptions for scene understanding, such as “I see a fruit stand with apples and oranges” or “I see a bakery with an ‘OPEN’ sign on its door.” Meanwhile, it continuously builds a memory from captured frames. Upon revisiting the market a week later, StateScribe compares the current view with stored memory and announces changes during exploration, e.g., “The fruit stand at your 10 o’clock is selling watermelon and guava; it sold apples and oranges last week,” “The bakery on your right now shows a ‘CLOSED’ sign; it was open at this time last week,” or “A construction site is at your 1 o’clock, 10 feet away.” Users can also query temporal changes via speech, e.g., “Has it blossomed now?”, receiving responses such as “Yes, the trees have blossomed; they had not last week.” Below, we describe how StateScribe achieves this experience with different system modules.

¹Code and datasets are available at <https://github.com/HumanAILab/StateScribe>

4.1 Memory Construction Module

In this module, StateScribe streams essential sensor data from the smartphone to the backend for processing and storage within a novel dual-layer memory architecture: (i) Episodic Scene Memory and (ii) Object-Centric Temporal Memory. These memory structures are informed by our formative study, which shows that observed changes primarily fall into three categories: single-object changes, spatial changes (e.g., multi-object rearrangement), and attribute changes within an object. However, this design may be limited in capturing fast motion or activity-based changes, which may require additional sensors and computational resources. We leave these as future work and discuss in Section 7.

Episodic Scene Memory (ESM). To capture the user’s visual context in real time, we design ESM to retain full observations within a configurable temporal context window (e.g., the last 10 days, 5 hours, or single visit), while archiving older data (Figure 2a). ESM serves as the primary memory retrieval source, storing data streamed from the mobile app to the backend at one data frame per second (FPS), including RGB-D frames, confidence maps, camera poses, and associated metadata. Upon receipt, the backend resizes images, refines the depth maps using confidence maps to filter out low-confidence pixels, generates visual embeddings, and asynchronously compresses and stores each instance as an *ESM data frame* on local disk. We adopt a frame-based representation over 3D point clouds because keyframes provide higher visual quality for VLM-based understanding and can later be compared directly for change detection (Section 4.2). Spatial information can be encoded efficiently as text using camera poses and 3D bounding boxes, enabling faster, more memory-efficient comparison and retrieval.

Object-centric Temporal Memory (OTM). To enable persistent tracking of key objects, we introduce OTM, an object-centric memory layer that maintains long-term object dynamics beyond ESM’s temporal window. Specifically, OTM maintains a collection of tracked real-world objects, each represented as a chronological sequence of discrete snapshots. Each snapshot captures the object’s state at a specific moment, including its status relative to the previous scene (e.g., appeared, disappeared, replaced), description, visual embedding, and a world-anchored 3D bounding box (Figure 3f,g). A new OTM entry is created when a change is detected for an object (e.g., it appears, disappears, or is replaced). When the changed object shares a similar 3D location with a prior OTM object, a new snapshot is appended to the existing OTM entry. These snapshots do not overwrite previous ones; instead, OTM preserves the full history of each tracked object. We detail this in the next section.

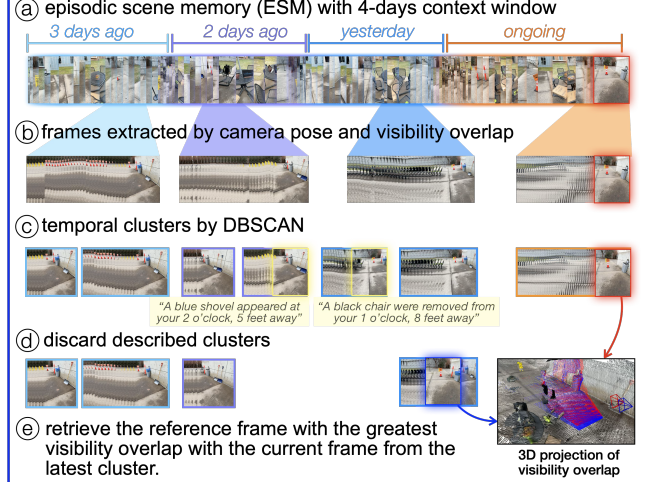
4.2 Change Detection and Memory Update Module

In this module, StateScribe identifies changes when a user revisits a location by comparing the current frame with its ESM and OTM constructed from prior visits. The process involves: (i) selecting a reference frame from ESM, (ii) comparing reference and current frames to detect changes, and (iii) updating OTM. We develop algorithms and integrate lightweight VLMs to balance accuracy and latency for live interactions (Section 4.7).

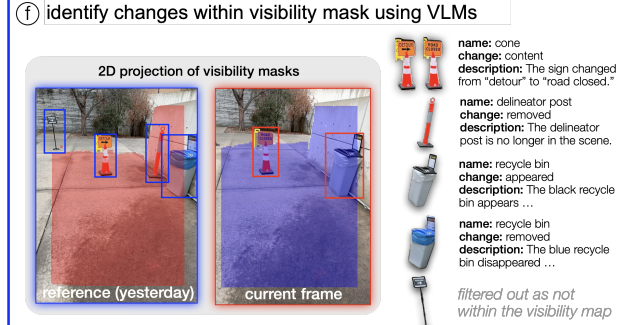
Step 1: Selecting a reference frame from ESM. The key challenge is to develop a scalable pipeline to efficiently identify the

Pipeline for change detection & memory update

step1: select a reference frame from ESM



step 2: compare reference and current frames for changes.



step 3: update object-centric temporal memory (OTM)

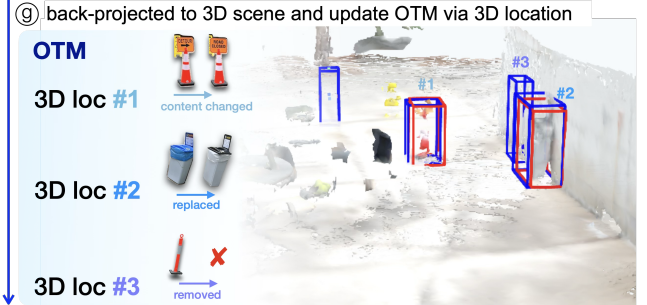


Figure 3: Change detection and memory update pipeline. (a) Given the current frame (rightmost, outlined in red), (b) StateScribe queries its ESM by retrieving past frames with similar camera pose and visibility. (c) These candidate frames are temporally clustered using DBSCAN. (d) Clusters whose changes have been announced are discarded. (e) From the most recent valid cluster, StateScribe selects the reference frame with the highest visibility overlap with the current frame. (f) A VLM compares the reference and current frames to detect changes, producing detailed metadata, such as object identity, change type, descriptions, and 2D/3D bounding boxes. (g) Finally, the detected changes are back-projected into the 3D space to update the OTM.

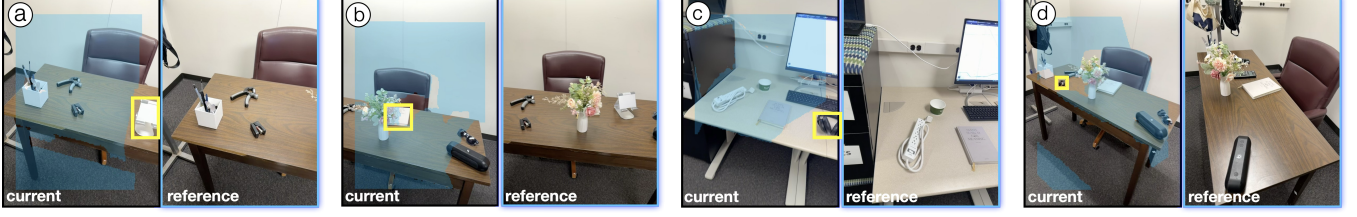


Figure 4: Examples of visibility-mask filtering of potential false positives. The (a) tablet stand, (b) book, (c) headphone, and (d) stapler were not detected as changes because they fell outside the visibility mask.

appropriate reference frame from the ESM as it grows over time. To address this, StateScribe utilizes a hierarchical frame retrieval technique. First, the current camera pose is used as a query to filter and retrieve the *ESM data frames* with similar poses, where the differences lie within translation (d_{thres}) and rotation (θ_{thres}) thresholds. Each candidate *ESM data frame* that passes the pose filter is then evaluated against the current camera pose using a bidirectional visibility score, $S_{overlap}$. Specifically, let P_r denote the set of valid 3D points back-projected from a reference *ESM data frame* r using its RGB-D data and camera intrinsics, where each $\mathbf{p} \in P_r$ represents a 3D point in the reference camera coordinate frame. Let $\pi_c(\cdot)$ denote the projection function that maps a 3D point from the reference camera coordinate to the current camera frame c . The resulting projection onto the current frame is referred to as the *visibility mask* (Figure 3f, right). We define the coverage ratio of the *visibility mask* over the current frame as:

$$o_{r \rightarrow c} = \frac{1}{|P_r|} \sum_{\mathbf{p} \in P_r} \mathbb{1}[\pi_c(\mathbf{p}) \in \Omega_c] \quad (1)$$

where Ω_c denotes the valid image domain of the current frame. Similarly, $o_{c \rightarrow r}$ is computed in the reverse direction. The final bidirectional visibility score is defined as the harmonic mean of the two directional coverage ratios:

$$S_{overlap} = \frac{2 o_{r \rightarrow c} o_{c \rightarrow r}}{o_{r \rightarrow c} + o_{c \rightarrow r}} \quad (2)$$

Second, candidates that satisfy the overlap thresholds are grouped into temporal clusters via DBSCAN [23] based on their timestamps (Figure 3c). Specifically, clustering is performed with a temporal radius of ϵ seconds and a minimum cluster size of N_c frames. StateScribe then scans the clusters in chronological order, discards those whose changes have already been announced, and selects the reference frame with the highest visibility score $S_{overlap}$ from the latest valid cluster. In our setup, we set $d_{thres} = 1.5$ meters, $\theta_{thres} = 40^\circ$, $\epsilon = 10$, and $N_c = 2$.

Step 2: Comparing reference and current frames to detect changes. Once a reference frame is selected, StateScribe identifies scene changes by prompting a VLM with both the reference and current frames to highlight changed objects using bounding boxes (Prompt #1). It returns the type of change for each detected object, including appeared, removed, or content changed, along with a textual description and a confidence score. StateScribe then rejects bounding boxes whose overlap with their corresponding visibility masks (described in step 1) falls below a threshold X (Figure 4). The remaining 2D bounding boxes are further filtered by confidence and size, then passed to a segmentation model to extract object masks and generate visual embeddings. These masks are subsequently projected into 3D space to estimate the objects' 3D

locations. Our current setup processes at 1 FPS, and we empirically set the temporal clustering parameters to $X = 0.45$.

Step 3: Updating OTM. Each detected change, along with its 2D mask, 3D bounding box, and the associated change type and description, is then updated in OTM. We determine if a detected object is new or corresponds to an existing OTM entry by computing the 3D Intersection over Union (IoU) between the detected 3D bounding box B_{new} and a prior 3D bounding box B_{prior} :

$$\text{IoU}(B_{new}, B_{prior}) = \frac{V_{int}}{V_{new} + V_{prior} - V_{int}} \quad (3)$$

where V_{int} denotes the intersection volume of the two boxes. StateScribe iterates over existing objects in OTM and retains candidate prior boxes B_{prior} whose IoU with B_{new} exceeds a minimum threshold γ . For the remaining candidates, StateScribe computes the cosine similarity between the visual embeddings of the new and prior objects. If the similarity exceeds a threshold Y , the object is treated as the same instance and recorded as a new snapshot in the corresponding OTM entry; otherwise, it is stored as a new OTM entry. Identified change snapshots, including change type, descriptions, reference, and current images, are then queued in the scheduling module to present (Section 4.5). We set $\gamma = 0.08$ and $Y = 0.7$.

4.3 Live Description Generation Module

In this module, StateScribe generates live visual descriptions to support scene understanding. Unlike prior systems that use multiple VLMs and adaptive prompting for context-aware descriptions [15, 18], StateScribe adopts a simplified approach using frame filtering and a single VLM with a fixed prompt (Prompt #2). To detect new scenes, it computes visual embeddings for incoming frames and compares them with the most recently described frame using cosine similarity; frames exceeding a threshold (τ_{visual}) are treated as redundant and skipped. To further reduce repetition, generated descriptions are also compared via text embeddings with prior outputs, and those exceeding a threshold (τ_{text}) are discarded. The remaining descriptions are queued for presentation (Section 4.5). We empirically set $\tau_{visual} = 0.85$ and $\tau_{text} = 0.80$ by default.

4.4 Interactive Q&A Module

In this module, StateScribe allows users to interrupt ongoing descriptions and issue queries. It supports change-related queries over single or multiple objects across time, within specified intervals, and their spatial variations. This is achieved by a Q&A agent that interprets each query and invokes tools over ESM, OTM, and spatial data (Prompt #4): (i) ESM retrieval for current scene queries, returning the N most recent frames; (ii) OTM retrieval for temporal

changes, returning recent object snapshots with distance and direction; and (iii) Spatial information retrieval for spatial queries, returning object locations (e.g., “11 o’clock, 5 feet away”). These tools are composed based on query intent. For example, “Where is the cone?” uses ESM retrieval and Spatial information retrieval, while “What changed since my last visit?” uses OTM retrieval.

4.5 Description Scheduling & Delivery Module

In this module, StateScribe coordinates change snapshots, live descriptions, and interactive Q&A. Visual queries are prioritized, after which live and change descriptions resume. StateScribe maintains a buffer ($N = 3$) of recent live descriptions and change snapshots. Live descriptions that persist beyond a time threshold (e.g., 6 seconds) are considered outdated and discarded, while change snapshots (e.g., *appeared*, *removed*, *content changed*), along with their images and 3D locations, are retained. Using this aggregated context in a single prompt (#3), StateScribe employs a VLM to infer higher-level events, such as object replacement (via overlapping 3D positions) or movement (via disappearance and reappearance across locations).

4.6 Mobile User Interface

StateScribe provides a mobile app interface (Figure 2a). The home screen displays visits as cards, each with a thumbnail and an AI-generated title. When returning to a location, users can select the corresponding card to open a live camera view for relocalization and resume StateScribe’s memory for that place. For new locations, they can create a visit using the “Add” button. Users can also rename or delete visits. While the current interface requires users to manually recall and select past visits, future iterations could incorporate GPS or indoor localization data to automatically cluster visits or create a new visit upon arrival at a new location.

4.7 Implementation Details

To enable StateScribe’s spatial and temporal memory capabilities, we use a smartphone as the primary device to access depth information, which is not available on the current smartglasses that typically rely on monocular cameras. We also adopt a low frame rate (i.e., 1 FPS) to support long-term use and scalability for memory storage, retrieval, and live interaction, in contrast to the higher frame rates used in prior systems [15, 18]. Our StateScribe mobile app is deployed on an iPhone 17 Pro and relies on camera poses and RGB-D images are continuously captured via Apple’s ARKit framework. These data streams are transmitted to the backend at one FPS over a TCP connection. StateScribe leverages ARKit’s built-in geometric relocalization to align the current camera coordinates with those from prior visits to the same location. Each location is assigned a unique identifier by the mobile application. Upon successful relocalization, the backend uses this identifier to load the corresponding ESM and OTM. The backend is hosted on a MacBook M4 Max for our user study, while the technical evaluation is conducted on a desktop with an AMD Ryzen Threadripper PRO 7965WX CPU and an NVIDIA RTX 6000 Ada GPU.

We adopt lightweight VLMs for live interactions to balance latency and accuracy. Specifically, live descriptions are generated using Gemini 3.1 Flash-Lite Preview, while change detection, Q&A, and buffered description summarization are handled by Gemini

3 Flash Preview. FastSAM [78] is used for 2D mask segmentation. Visual embeddings are extracted using a pre-trained DINOv3 model [64], and text embeddings are generated with all-MiniLM-L12-v2. Finally, we limited the ESM context window to the previous visit for both user and technical evaluations.

5 Technical Evaluation

We evaluated whether StateScribe (i) preserves longitudinal memory across visits, (ii) scales efficiently with growing data, and (iii) improves real-time change awareness in 3D environments. We focused on the live description pipeline—change detection, memory updates, and announcements—and excluded the auxiliary Q&A module because participants’ questions varied and were not always change-related. Evaluating the Q&A module remains future work.

5.1 Datasets

To simulate extended use, we constructed a dataset of repeated visits to the same environments over time. The dataset includes three environments (Figure 5): (i) a shared office with varying objects, (ii) a grocery store with densely arranged items across shelves, and (iii) an outdoor courtyard featuring signs and barriers. Each scenario contains 11 recorded visits (around 2–4 minutes per visit), including a video, RGB-D frames, camera intrinsics, and poses, capturing observations at different time points. We systematically introduced object changes based on the spatial layout of the shared office, 104 for the grocery, and 83 for the outdoor scenario. In total, the dataset contains 291 annotated changes, labeled by three researchers with 3D world locations, object identities, and change categories.



Figure 5: Our study environments, including a shared office, a simulated grocery store, and an outdoor courtyard.

5.2 Baseline Conditions and Evaluation Metrics

To evaluate the effectiveness of StateScribe’s design, we compared it against two VLM-only baselines: a *Live model* and an *Offline video model*. For both baselines, we concatenated the recorded videos from multiple visits in the same scenario and sampled them at 1 FPS. In the *Live model*, we prompted the VLM (i.e., gemini-3-flash-preview) at 1 FPS to approximate real-time interaction similar to StateScribe (Prompt #6). This baseline examines whether StateScribe improves over a direct VLM-only real-time pipeline. In the *Offline video model* condition, we provided the concatenated 1 FPS video to gemini-3.1-pro as a single input for detecting changes within or across visits (Prompt #5). This baseline evaluates StateScribe against an offline setting that processes all observations jointly rather than sequentially. For both baselines, we used the same Gemini model family as in StateScribe to isolate the effect of system design rather than differences across model families.

We used Gemini 3.1 Pro as an LLM judge to compare predicted objects, locations, change types, and descriptions (e.g., “The white Milano cookie box has been removed from the shelf.”) with ground

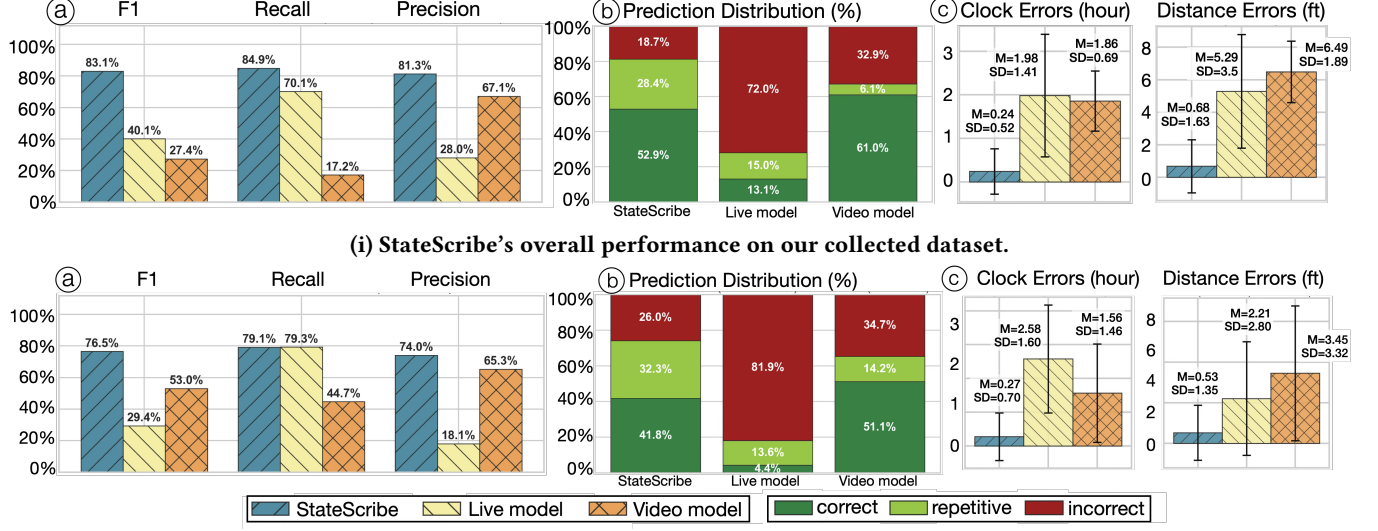


Figure 6: StateScribe's overall performance across two evaluation settings: (i) technical evaluation with our collected datasets and (ii) user study with data collected by nine BLV participants. Both use the same evaluation metrics: (a) performance metrics including *F1 score*, *recall*, and *precision* across models; (b) *prediction distribution* for StateScribe, Live model, and Offline video model, showing the number of correct, repetitive, and incorrect outputs; and (c) error analysis showing mean and standard deviation for *Clock* and *Distance Errors*.

truth. The judge assessed whether each description captured the ground-truth event, rather than its standalone quality. Nevertheless, this design may not fully eliminate same-family bias, although a preliminary cross-family validation using GPT-5.5 (rather than a Gemini-family model) as the judge yielded similar trends. Future work could further validate these findings through additional human ratings. We reported *precision*, *recall*, and F_1 -score, and analyzed output counts for *repetitiveness* and *coverage* (Section 5.3.1). Spatial accuracy, including *clock* and *distance* errors, was computed directly from annotated 3D boxes, camera poses, and ground truth without the LLM judge. We also repeated this evaluation on data from BLV participants (Section 5.3.2). To assess scalability, we simulated 110 visits by repeating an 11-visit sequence 10 times and measured *storage* and *latency* (Section 5.3.3).

5.3 Results

We first report the overall performance across models on our collected dataset (Figure 6.i) and on data captured by BLV participants in our user study (Figure 6.ii). We then evaluate the performance of StateScribe under extended use (Figure 7).

5.3.1 Overall Performance. We reported the overall performance across all 291 ground-truth instances collected from three scenarios, each with 11 visits. StateScribe outperformed both baselines in change detection (Figure 6.i-a). It achieved an F_1 -score of **83.1%**, substantially higher than the *Live model* (40.1%) and the *Office video model* (27.4%). This performance was driven by both high *recall* (84.9%) and *precision* (81.3%), indicating that StateScribe captures most changes while avoiding excessive false positives. In contrast, the *Live model* achieved moderate *recall* (70.1%) but low *precision* (28.0%), resulting in many incorrect predictions. The *Offline video model*, while achieving moderate *precision* (67.1%), exhibited low *recall* (17.2%), missing detecting most changes that lead to poor overall performance. Most false positives in both models occurred

when objects appeared near the image edge without prior visibility (Figure 9). Likewise, objects misclassified as *removed* had merely left the camera's view while still existing (Figure 9), whereas StateScribe mitigated this using visibility masks (Figure 4). Errors in StateScribe mainly arose for distant objects beyond LiDAR coverage, text cutoffs or light reflections (Figure 8).

When examining the *Prediction Distribution* across models, including counts of correct, incorrect, and repetitive predictions, we observed a wide variance in output behavior (Figure 6.i-b). StateScribe produced 444 predictions, including 81.13% correct ($N=361$, including 126 repetitive), and 18.7% incorrect outputs ($N=82$). In contrast, the *Live model* produced a large number of predictions ($N=1558$), despite only 291 true changes in the dataset. As a result, the majority of its outputs were incorrect (72.0%, $N=1121$), while 28.1% were correct ($N=437$, including 233 repetitive), reflecting both a high error rate and considerable redundancy. Furthermore, the *Offline video model* generated far fewer predictions ($N=82$), comprising 61% correct ($N=50$, including 5 repetitive), and 32.9% incorrect outputs ($N=27$). This aligns with its previously observed high *precision* and low *recall*, suggesting conservative behavior or limited ability to capture all changes.

In terms of spatial accuracy (Figure 6.i-c), StateScribe outperformed both models, achieving lower clock-direction errors (in hour units; $M=0.24$, $SD=0.52$) than the *Live model* ($M=1.98$, $SD=1.41$) and the *Offline video model* ($M=1.86$, $SD=0.69$). A similar trend was observed for distance estimation: StateScribe yielded smaller errors (in feet; $M=0.68$, $SD=1.63$) than the *Live model* ($M=5.29$, $SD=3.50$) and the *Offline video model* ($M=6.49$, $SD=1.89$).

Overall, StateScribe achieved a better balance between coverage and accuracy, identifying more changes than both models while having substantially fewer errors and repetitive outputs than the *Live model*. This trend also extended to spatial accuracy, with StateScribe showing lower clock-direction and distance estimation errors.

5.3.2 StateScribe’s performance in user study. Next, we used the same evaluation metrics on the data collected from our user study (see Section 6.1 for data details), where camera frames from BLV participants may differ due to camera aiming. Overall, StateScribe showed similar trends (Figure 6.ii), outperforming both baselines across 269 tasks. It achieved an F_1 -score of **76.5%**, higher than the *Live model* (29.4%) and the *Offline video model* (53.0%). This performance is driven by both high *recall* (79.1%) and *precision* (74.0%), indicating effective change detection with limited false positives. In comparison, the *Live model* achieved similar *recall* (79.3%) but very low *precision* (18.1%), leading to many false positives. The *Offline video model* showed moderate *precision* (65.3%) but low *recall* (44.7%), missing many changes. Similar trends were observed in spatial accuracy. StateScribe achieved lower clock-direction error (in hours; $M=0.27$, $SD=0.70$) than the *Live model* ($M=2.58$, $SD=1.60$) and the *Offline video model* ($M=1.56$, $SD=1.46$). It produced smaller distance errors (in feet; $M=0.53$, $SD=1.35$) compared to the *Live model* ($M=2.21$, $SD=2.80$) and the *Offline video model* ($M=3.45$, $SD=3.32$). Post-hoc analysis suggests that the performance difference between datasets collected by sighted and BLV participants (Figure 6i-a & ii-a) stems from movement patterns. Sighted participants tended to move faster, whereas BLV participants lingered longer in each scene despite less stable camera handling, giving the video model more opportunities to detect changes and yielding higher recall and F_1 scores. Nevertheless, StateScribe outperformed the baseline models on both datasets.

5.3.3 StateScribe’s latency and memory footprint over extended use. To model long-term use and measure latency and memory growth, we repeated 11 visits 10 times (110 visits per scenario), producing 17,350 frames for Office, 17,460 frames for Grocery, and 18,360 frames for Outdoor. Each scenario corresponds to about 5 hours at 1 FPS. This setup is valid because StateScribe’s context window included only the prior visit (Section 4.7), so every visit triggered the full change-detection and memory-update pipeline.

In terms of *latency*, it remains low across all scenarios, including Grocery (Median=0.37s, Mean=1.42s), Office (Median=0.21s, Mean=1.32s), and Outdoor (Median=0.14s, Mean=1.11s), which reflected long-tailed distributions with occasional outliers. This indicated that most interactions were processed quickly, with only rare spikes. To further examine latency contributions, we decomposed processing time into system components. Across all scenarios, *VLM inference* dominated latency (Grocery: $M=1.02$ s; Office: $M=1.08$ s; Outdoor: $M=0.91$ s). In contrast, *frame queuing* for processing ($M=0.11$ s, 0.10s, 0.09s), *reference matching* for retrieving past frame data ($M=0.27$ s, 0.12s, 0.07s), and *post-processing* for updating OTM ($M=0.01$ s across all scenarios) contributed relatively minor overhead. Notably, reference matching was higher in Grocery ($M=0.27$ s) than in Office ($M=0.12$ s) and Outdoor ($M=0.07$ s), likely due to the denser scenes increasing matching complexity. Overall, these results suggest that StateScribe maintains stable performance over extended use, with latency primarily bounded by VLM inference rather than system-level overhead.

Over extended use for 110 visits, StateScribe maintained a bounded memory footprint that scales proportionally with the number of visits. The final OTM size across 110 visits was 49.4 MB (Grocery),

54.6 MB (Office), and 39.4 MB (Outdoor). The Office scenario exhibited the largest memory due to more change snapshots ($N=1,517$) of tracked objects ($N=204$), while Grocery showed moderate growth with moderate snapshots ($N=1,107$) of tracked objects ($N=192$), and Outdoor remained smaller due to fewer snapshots ($N=999$) of tracked objects ($N=174$). Overall, latency did not increase with OTM size, as our memory architecture reads only prior-visit data from ESM, discards outdated information, and stores only essential object changes in OTM. The current OTM scale (110 visits) also remains computationally manageable during retrieval.

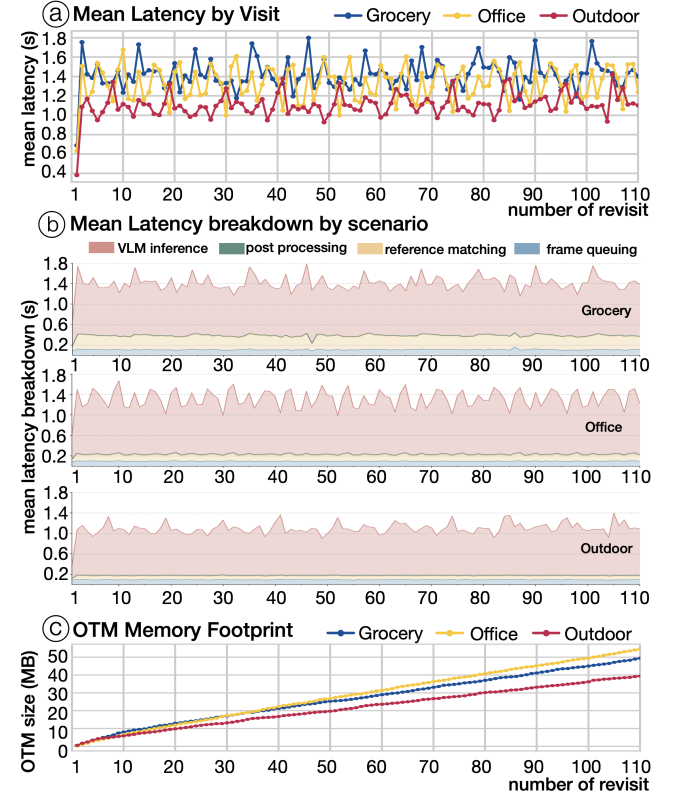


Figure 7: StateScribe’s (a) mean latency, (b) their breakdown across system components, and (c) memory footprint over 110 visits in three scenarios: Grocery, Office, and Outdoor.

6 User Study

Next, we conducted a user study with 9 BLV participants to examine: (i) How effectively does StateScribe support users in perceiving and understanding changes in different environments? (ii) How do participants perceive the user experience of StateScribe?

Participants, 6 Male and 3 Female, were recruited through prior contacts and snowball sampling (Table 2). They aged 25–74 (mean: 49.7); 6 identified as blind and 3 as having low vision. Most had prior experience with RSA services (e.g., Be My Eyes [3], Aira [2]) and AI tools (e.g., Seeing AI [6], Be My AI [1]).

6.1 Procedure, Tasks and Analysis

The study consisted of three sessions: (i) an *onboarding session*, where participants learned to use StateScribe; (ii) a *main task session*, where participants were asked to identify changes in three different

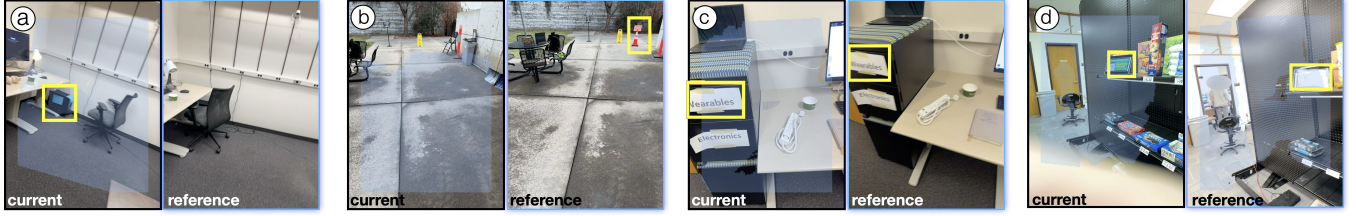


Figure 8: StateScribe errors. (a) Objects that are present but occluded before may be incorrectly classified as “appeared.” (b) The visibility mask may be truncated for regions outside the LiDAR coverage. (c) Truncated text may be misinterpreted as an attribute change, such as from “Wearables” to “Nearables.” (d) Light reflections may also be mistaken for attribute changes.

scenarios, including a shared office, a grocery store, and an outdoor courtyard; and (iii) an *interview session*, where participants reflected on their experiences and responded to Likert-scale questions.

For each scenario in the *main task session*, participants first explored the environment and then revisited it. During each visit, researchers introduced *within-visit changes* by moving objects in real time to simulate common dynamics in shared or public spaces (e.g., placing or removing items; Table 3). Examples included placing a bottle on a shared table, taking items in the grocery store, or removing outdoor obstacles by workers (Table 3). Between visits, the researcher introduced *cross-visit changes* to simulate longer-term rearrangements (e.g., days or weeks). These included moving chairs in the shared office, price changes in the grocery store, and removing signage in the outdoor walkway. These changes approximated everyday dynamics within the time constraint of a two-hour study. In total, the shared office and grocery store scenarios each included 13 changes (8 within-visit, 5 cross-visit), while the outdoor scenario included 12 changes (8 within-visit, 4 cross-visit).

Participants independently explored the indoor environments, including a shared office and a grocery store. In the outdoor scenario, a researcher accompanied participants for safety but did not interfere with their exploration. In total, we collected 18 visit records for the shared office (117 tasks = 9 participants × 13 changes), 16 for the grocery store (104 tasks = 8 participants × 13 changes; P9 excluded due to corrupted data), and 8 for the outdoor courtyard (48 tasks = 4 participants × 12 changes), as only four participants completed the outdoor condition due to varying weather conditions. In total, this yielded 269 tasks. During each visit, participants thought aloud as they identified changes. We evaluated task completion, Likert-scale ratings, and qualitative feedback, marking tasks as successful when changes were accurately described.

6.2 Results

6.2.1 How effectively does StateScribe support users in perceiving and understanding changes in different environments? Overall, participants successfully completed most tasks within around 10 minutes, achieving completion rates of 82.9% (shared office), 82.7% (grocery), and 89.6% (outdoor). In the shared office scenario, issues were mainly due to camera aiming: some regions (e.g., parts of desks) were not captured, leading to incomplete memory, and subtle view changes, such as slightly pulled-out chairs, were hard to detect (4/9 misses). In the grocery scenario, removing the cart from the aisle had a lower completion rate (50%) than in other tasks, as participants focused the camera on nearby shelf items, and often missed objects at the periphery of the scene. Consequently, the shopping cart, positioned near the edge of the scene,

was often partially or not captured at all. Additionally, sparkling water boxes on the upper shelves introduced visual ambiguity, leading to missed detections of Lay’s chips (3/8 misses). In contrast, the outdoor scenario yielded higher performance, as the larger space and sparser object layout made changes easier to capture and detect.

6.2.2 How do participants perceive the user experience of StateScribe? Participants rated StateScribe as easy to learn and use ($M=5.8$, $SD=0.8$), noting that automatic descriptions and the hold-to-ask interaction were intuitive. However, some found manually retrieving prior visits cumbersome, suggesting solutions by “using GPS and auto-relocalization when the camera is always on, like smartglasses” (P1). Participants perceived the descriptions as accurate ($M=6.0$, $SD=1.0$) and expressed high trust in StateScribe ($M=5.7$, $SD=1.4$), supported by temporal and directional details (also evidenced by our technical evaluation at Section 5.3.2). For instance, P8 noted “it was cool how it could say 7 minutes ago was this, 5 minutes ago was that,” while P3 emphasized that “exact directionality makes me confident to trust the system.” These perceptions were corroborated by low-vision users (P1, P6, P8), who could verify details at close range, and blind users, who found descriptions consistent over time (e.g., P4: “when it matches what it said before, that gives me confidence that it’s accurate.”). Although reported reasonable coverage of the objects of interest ($M=5.2$, $SD=1.6$), participants wanted richer, more personalized interactions. For example, P4 wanted descriptions “narrowly focused if only interested in chips,” while P1 preferred a more human-like tone, such as “You were here two days ago and did [X], and [Y] was present at that time.”

Participants were impressed by StateScribe’s memory capabilities, which are absent from current off-the-shelf visual assistive technologies. For example, P5 viewed it as a “promising integration with Seeing AI scan function,” while P3 felt “astounded at how well it remembers and describes differences as you move. That’s not something Be My AI can do.” Participants also found the descriptions helpful ($M=5.9$, $SD=1.2$) and willing to use StateScribe in real-world settings ($M=5.6$, $SD=1.4$). For example, P8 described a bus stop navigation experience that StateScribe could be helpful: “Maps told me to stand in one place to catch the bus today, but it was relocated by a couple of hundred feet, so I had to walk around to find it. This [StateScribe] could help me find the relocation out more quickly.” P2, who was a mental health counselor, suggested “it could be helpful to know the person’s attire, like in military manner or slovenly from time to time, and if their facial expression changes during conversation.” Despite positive feedback, some reported information overload and wanted greater control, especially when using other aids. For example, P6 suggested omitting changes, like “A chair appeared in front of you. It was not there before,” when already perceived with a cane.

7 Discussion and Future Work

We discuss our lessons learned and design implications for extending StateScribe to broader use cases in the long term.

Generalization to real-world use. StateScribe’s technical core is the pipeline that detects changes, constructs, and updates memory with an always-on camera. In scenarios that benefit from continuous change detection, such as tracking price changes, locating objects moved within shared spaces, or identifying changes to commute routes, BLV users could keep StateScribe active on their devices throughout the activity. Since the current pipeline relies on RGB-D images and camera poses, it could be extended to wearable devices, such as smart glasses, with comparable sensing capabilities. Integrating GPS could also reduce the need to manually select prior visits by enabling automatic relocalization, supporting more lightweight, hands-free interaction. More broadly, this pipeline could serve as a foundation for other assistive technologies. For instance, RSA services rely on real-time human communication via live video, while emerging systems [4, 7] demonstrate the promise of live human-AI interactions. However, these systems often lack persistent memory: sighted assistants may change across sessions [48], while live-video AI systems generally do not maintain persistent spatial representations [17]. We therefore envision integrating StateScribe’s change detection and memory capabilities into these systems, activating them whenever a live-video session begins to enhance the ability of sighted assistants and AI systems to answer questions about long-term environmental changes.

Supporting more types of changes. As noted in Section 4.1, StateScribe focuses on object-level changes, enabling effective real-time detection and across revisits. However, at 1 FPS, it may miss fast or transient changes (e.g., moving cars, flash lights, human activities). Although increasing frame rate could help, it may introduce trade-offs in computation, storage, and retrieval. An alternative is to augment StateScribe with additional sensors. For instance, event cameras can capture high-frequency motion and light changes asynchronously, while infrared cameras can detect otherwise invisible changes such as temperature variations (e.g., heating or cooling objects). Integrating these sensors could expand the range of detectable changes and improve robustness by cross-referencing.

Integrating different information cursors to deliver the right information timely. StateScribe uses the user’s 3D location (from camera pose) as an information cursor, a mechanism to indicate information of interest, to retrieve memory, and detect changes across revisits. However, our study showed that user needs extended beyond change detection and vary by context, and StateScribe does not always surface the most relevant information in time. For example, participants often held items and sought visual details (e.g., color or flavor) in grocery settings, or needed detailed information (e.g., whether stairs go up or down) and signage texts in outdoor settings. These needs could be better supported by prior systems that use different user activities as information cursors, such as leveraging hand-object interactions as intent cues [43, 44, 51], or using camera motion to obtain adaptive visual details [15]. Together, these works shift visual assistance from reactive question answering to proactive interaction. However, effectively integrating multiple information cursors and delivering the right information at the right time remains an open challenge [14]. Future work should explore

how to model user context, select appropriate information cursors, and map user intent to timely descriptions.

Towards a long-term assistive AI companion. StateScribe represents an initial step toward memory architectures grounded in the physical world for real-time change detection and announcement, complementing prior work on text-based memory for LLM-based agents [5, 57, 58]. It outperforms both a VLM-only *Live model* and an *Offline video model* (Section 5.3.1), while supporting extended use with low latency and memory overhead (Section 5.3.3). However, challenges remain before such systems can serve as reliable long-term assistive companions. Participants expected StateScribe to account for their assistive context (e.g., the use of a white cane) and evolving intent (e.g., tracking interested objects or locations). They also wanted richer contextual reminders when revisiting locations, similar to those a human companion might provide (Section 6.2). Beyond object-level changes, memory could encode an environment’s semantic and spatial structure, including the store layouts and aisle organization envisioned by our participants. Such capabilities could be supported by incorporating structured scene memory as an additional representational layer [8, 9, 25, 34, 53, 56, 61, 79], as well as active vision that guides what the system inspects, revisits, or remembers [10, 24, 52]. These memories could also be augmented with community-contributed data accumulated over time, such as user-uploaded photos and videos, satellite imagery, and street-level maps [27, 35, 37, 38, 49]. More broadly, realizing this vision will require sustained community efforts in dataset and system development [65, 66], model training, and longitudinal evaluation to better align long-term assistive AI systems with users’ evolving needs.

8 Conclusion

In this paper, we introduced StateScribe, a system that enables accessible awareness of real-world changes across revisits. Informed by a formative study, we identified meaningful change categories for BLV people and developed a dual-layer memory architecture comprising Episodic Scene Memory and Object-centric Temporal Memory. This architecture supports real-time change detection, persistent tracking across revisits, and enhanced scene understanding through accurate spatial descriptions, live descriptions, and queries over current and past changes. Our technical evaluation shows that StateScribe detected changes more accurately than the VLM-only live model or offline video model while remaining low-latency and memory-efficient for extended use. A user study further showed that StateScribe helped BLV people effectively identify changes across revisits in different locations. Through this work, we also recognized that enabling an AI-assisted companion requires broader observation of changes, reasoning over those changes, and adapting information to individual users, intents, and contexts.

Acknowledgments

This research was supported by Samsung Research America, a Google Academic Research Award, and the National Science Foundation Award No. 2516629. Ruei-Che Chang was also supported by the Apple Scholars in AI/ML PhD Fellowship. We thank Prof. Pei Zhang for sharing lab space and Jiale Zhang for coordination. We also thank Cole Gleason and Project Mutatio for early insights.

References

- [1] 2023. Introducing Be My AI (formerly Virtual Volunteer) for People who are Blind or Have Low Vision, Powered by OpenAI's GPT-4. <https://www.bemyeyes.com/blog/introducing-be-my-eyes-virtual-volunteer>
- [2] 2026. Aira. <https://aira.io/>
- [3] 2026. BeMyEyes. <https://www.bemyeyes.com/>
- [4] 2026. Gemini Live. <https://gemini.google/overview/gemini-live/>
- [5] 2026. Memory and new controls for ChatGPT. <https://openai.com/index/memory-and-new-controls-for-chatgpt/>
- [6] 2026. SeeingAI. <https://www.seeingai.com/>
- [7] 2026. Voice with real-time video in ChatGPT. <https://chatgpt.com/features/voice-with-video/>
- [8] Muhammad Qasim Ali, Saeed Nair, Alexander Wong, Yuchen Cui, and Yuhao Chen. 2025. GraphPad: Inference-Time 3D Scene Graph Updates for Embodied Question Answering. *arXiv preprint arXiv:2506.01174* (2025).
- [9] Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. 2025. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2838–2845.
- [10] Ruzena Bajcsy. 1988. Active perception. *Proc. IEEE* 76, 8 (1988), 966–1005.
- [11] Samantha W. T. Chan. 2020. Biosignal-Sensitive Memory Improvement and Support Systems. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–7. doi:10.1145/3334480.3375031
- [12] Samantha W. T. Chan, Shardul Sapkota, Rebecca Mathews, Haimo Zhang, and Suranga Nanayakkara. 2020. Prompto: Investigating Receptivity to Prompts Based on Cognitive Load from Memory Training Conversational Agent. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 121 (Dec. 2020), 23 pages. doi:10.1145/3432190
- [13] Samantha W. T. Chan, Haimo Zhang, and Suranga Nanayakkara. 2019. Prospero: A Personal Wearable Memory Coach. In *Proceedings of the 10th Augmented Human International Conference 2019* (Reims, France) (AH2019). Association for Computing Machinery, New York, NY, USA, Article 26, 5 pages. doi:10.1145/3311823.3311870
- [14] Ruei-Che Chang. 2025. Enabling Real-World Assistive Agents: From Live Vision to Proactive Context-Aware Information Delivery. In *Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25)*. Association for Computing Machinery, New York, NY, USA, Article 2, 5 pages. doi:10.1145/3746058.3758468
- [15] Ruei-Che Chang, Yuxuan Liu, and Anhong Guo. 2024. WorldScribe: Towards Context-Aware Live Visual Descriptions. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 140, 18 pages. doi:10.1145/3654777.3676375
- [16] Ruei-Che Chang, Yuxuan Liu, Lotus Zhang, and Anhong Guo. 2024. EditScribe: Non-Visual Image Editing with Natural Language Verification Loops. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (St. John's, NL, Canada) (ASSETS '24). Association for Computing Machinery, New York, NY, USA, Article 65, 19 pages. doi:10.1145/3663548.3675599
- [17] Ruei-Che Chang, Rosiana Natalie, Wenqian Xu, Jovan Zheng Feng Yap, and Anhong Guo. 2025. Probing the Gaps in ChatGPT's Live Video Chat for Real-World Assistance for People who are Blind or Visually Impaired. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '25). Association for Computing Machinery, New York, NY, USA, Article 71, 14 pages. doi:10.1145/3663547.3746319
- [18] Ruei-Che Chang, Rosiana Natalie, Wenqian Xu, Jovan Zheng Feng Yap, Tiange Luo, Venkatesh Potluri, and Anhong Guo. 2026. TouchScribe: Augmenting Non-Visual Hand-Object Interactions with Automated Live Visual Descriptions. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (CHI '26). Association for Computing Machinery, New York, NY, USA, Article 747, 18 pages. doi:10.1145/3772318.3791308
- [19] Chufeng Chen, Michael Oakes, and John Tait. 2006. Browsing personal images using episodic memory (time+ location). In *Advances in Information Retrieval: 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006. Proceedings* 28. Springer, 362–372.
- [20] Matthew Cooper, Jonathan Foote, Andreas Girsensohn, and Lynn Wilcox. 2005. Temporal event clustering for digital photo collections. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 3 (Aug. 2005), 269–288. doi:10.1145/1083314.1083317
- [21] Laura Cushley, Neil Galway, and Tunde Peto. 2023. The unseen barriers of the built environment: navigation for people with visual impairment. *Town planning review* 94, 1 (2023), 11–35.
- [22] Fatma El-Zahraa El-Taher, Luis Miralles-Pechuán, Jane Courtney, Kristina Millar, Chantelle Smith, and Susan McKeever. 2023. A survey on outdoor navigation applications for people with visual impairments. *IEEE Access* 11 (2023), 14647–14666.
- [23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) (KDD '96). AAAI Press, 226–231.
- [24] Lei Fan, Mingfu Liang, Yunxuan Li, Gang Hua, and Ying Wu. 2024. Evidential active recognition: Intelligent and prudent open-world embodied perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16351–16361.
- [25] Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Ruijie Wu, Xi Chen, and Qing Li. 2025. Embodied VideoAgent: Persistent Memory from Egocentric Videos and Embodied Sensors Enables Dynamic Scene Understanding. *arXiv:2501.00358 [cs.CV]* <https://arxiv.org/abs/2501.00358>
- [26] Maxwell Forbes, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie. 2019. Neural naturalist: Generating fine-grained image comparisons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 708–717.
- [27] Jon E. Froehlich, Alexander J. Fiannaca, Nimer M Jaber, Victor Tsaran, and Shaun K. Kane. 2025. StreetViewAI: Making Street View Accessible Using Context-Aware Multimodal AI. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 43, 22 pages. doi:10.1145/3746059.3747756
- [28] Jim Gemmell, Lyndsay Williams, Ken Wood, Roger Lueder, and Gordon Bell. 2004. Passive capture and ensuing issues for a personal lifetime store. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences* (New York, New York, USA) (CARPE '04). Association for Computing Machinery, New York, NY, USA, 48–55. doi:10.1145/1026653.1026660
- [29] Rúben Gouveia and Evangelos Karapanos. 2013. Footprint tracker: supporting diary studies with lifelogging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (CHI '13). Association for Computing Machinery, New York, NY, USA, 2921–2930. doi:10.1145/2470654.2481405
- [30] Gillian R Hayes, Shwetak N Patel, Khai N Truong, Giovanni Iachello, Julie A Kientz, Rob Farmer, and Gregory D Abowd. 2004. The personal audio loop: Designing a ubiquitous audio-based memory aid. In *International Conference on Mobile Human-Computer Interaction*. Springer, 168–179.
- [31] Jennifer Healey and Rosalind W Picard. 1998. Startlecam: A cybernetic wearable camera. In *Digest of Papers. Second International Symposium on Wearable Computers* (Cat. No. 98EX215). IEEE, 42–49.
- [32] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. 2006. SenseCam: A retrospective memory aid. In *UbiComp 2006: Ubiquitous Computing: 8th International Conference, UbiComp 2006 Orange County, CA, USA, September 17-21, 2006 Proceedings* 8. Springer, 177–193.
- [33] Tetsuro Hori and Kiyoharu Aizawa. 2003. Context-based video retrieval system for the life-log applications. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval* (Berkeley, California) (MIR '03). Association for Computing Machinery, New York, NY, USA, 31–38. doi:10.1145/973264.973270
- [34] Wenbo Hu, Yining Hong, Yanjun Wang, Leison Gao, Zibu Wei, Xingcheng Yao, Nanyun Peng, Yonatan Bitton, Idan Szepkter, and Kai-Wei Chang. 2025. 3DLLM-Mem: Long-Term Spatial-Temporal Memory for Embodied 3D Large Language Model. *arXiv:2505.22657 [cs.CV]* <https://arxiv.org/abs/2505.22657>
- [35] William Huang, Xia Su, Jon E. Froehlich, and Yang Zhang. 2025. Accessibility Scout: Personalized Accessibility Scans of Built Environments. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 183, 18 pages. doi:10.1145/3746059.3747624
- [36] Mina Huh, Yi-Hao Peng, and Amy Pavel. 2023. GenAssist: Making Image Generation Accessible. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 38, 17 pages. doi:10.1145/3586183.3606735
- [37] Jared Hwang, John S. O'Meara, Zeyu Wang, Jasmine Zhang, and Jon E. Froehlich. 2026. BikeButler: A Personalized, Context-sensitive Bike Routing Tool using Open Data and VLM-based Analyses of Street View Imagery. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (CHI '26). Association for Computing Machinery, New York, NY, USA, Article 1069, 21 pages. doi:10.1145/3772318.3791292
- [38] Gaurav Jain, Leah Findlater, and Cole Gleason. 2026. SceneScout: Towards AI-Driven Access to Street Level Imagery for Blind Users. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (CHI '26). Association for Computing Machinery, New York, NY, USA, Article 1415, 22 pages. doi:10.1145/3772318.3790449
- [39] Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4024–4034.
- [40] Shiqi Jiang, Zhenjiang Li, Pengfei Zhou, and Mo Li. 2019. Memento: An Emotion-driven Lifelogging System with Wearables. *ACM Trans. Sen. Netw.* 15, 1, Article 8 (Jan. 2019), 23 pages. doi:10.1145/3281630

- [41] Doaa Khattab, Julie Buelow, and Donna Saccuteli. 2015. Understanding the barriers: Grocery stores and visually impaired shoppers. *Journal of accessibility and design for all: JACCES* 5, 2 (2015), 157–173.
- [42] Mik Lamming, Peter Brown, Kathleen Carter, Margery Eldridge, Mike Flynn, Gifford Louie, Peter Robinson, and Abigail Sellen. 1994. The design of a human memory prosthesis. *Comput. J.* 37, 3 (1994), 153–163.
- [43] Kyungjun Lee, Jonggi Hong, Simone Pimento, Ebrima Jarjue, and Hernisa Kacorri. 2019. Revisiting Blind Photography in the Context of Teachable Object Recognizers. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 83–95. doi:10.1145/3308561.3353799
- [44] Kyungjun Lee and Hernisa Kacorri. 2019. Hands Holding Clues for Object Recognition in Teachable Machines. 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. doi:10.1145/3290605.3300566
- [45] Kyungyeon Lee, Sohyeon Park, and Uran Oh. 2021. Designing Product Descriptions for Supporting Independent Grocery Shopping of People with Visual Impairments. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI EA '21). Association for Computing Machinery, New York, NY, USA, Article 425, 6 pages. doi:10.1145/3411763.3451806
- [46] Matthew L. Lee and Anind K. Dey. 2007. Providing good memory cues for people with episodic memory impairment. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility* (Tempe, Arizona, USA) (ASSETS '07). Association for Computing Machinery, New York, NY, USA, 131–138. doi:10.1145/1296843.1296867
- [47] Matthew L. Lee and Anind K. Dey. 2008. Lifelogging memory appliance for people with episodic memory impairment. In *Proceedings of the 10th International Conference on Ubiquitous Computing* (Seoul, Korea) (UbiComp '08). Association for Computing Machinery, New York, NY, USA, 44–53. doi:10.1145/1409635.1409643
- [48] Sooyeon Lee, Madison Reddie, Krish Gurdasani, Xiyang Wang, Jordan Beck, Mary Beth Rosson, and John M Carroll. 2018. Conversations for Vision: Remote Sighted Assistants Helping People with Visual Impairments. *arXiv preprint arXiv:1812.00148* (2018).
- [49] Chu Li, Rock Yuren Pang, Arnavi Chheda-Kothary, Ather Sharif, Henok Assalif, Jeffrey Heer, and Jon E. Froehlich. 2026. GeoVisA11y: An AI-based Geovisualization Question-Answering System for Screen-Reader Users. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (CHI '26). Association for Computing Machinery, New York, NY, USA, Article 463, 17 pages. doi:10.1145/3772318.3790334
- [50] Jiahao Nick Li, Zhuohao Jerry Zhang, and Jiaju Ma. 2024. OmniQuery: Contextually Augmenting Captured Multimodal Memory to Enable Personal Question Answering. *arXiv:2409.08250 [cs.HC]* <https://arxiv.org/abs/2409.08250>
- [51] Jongho Lim, Yongjae Yoo, Hanseul Cho, and Seungmoon Choi. 2019. TouchPhoto: Enabling Independent Picture Taking and Understanding for Visually-Impaired Users. In *2019 International Conference on Multimodal Interaction* (Suzhou, China) (ICMI '19). Association for Computing Machinery, New York, NY, USA, 124–134. doi:10.1145/3340555.3353728
- [52] Zhenyang Liu, Yongchong Gu, Yikai Wang, Xiangyang Xue, and Yanwei Fu. 2026. ActiveVLA: Injecting Active Perception into Vision-Language-Action Models for Precise 3D Robotic Manipulation. *arXiv preprint arXiv:2601.08325* (2026).
- [53] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Venamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Aleksandr Maksymets, Sergio Arnaud, et al. 2024. Openqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16488–16498.
- [54] Steve Mann. 1998. "WearCam"(The wearable camera): personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis. In *Digest of Papers. Second International Symposium on Wearable Computers* (Cat. No. 98EX215). IEEE, 124–131.
- [55] Steve Mann, James Fung, Chris Aimone, Anurag Sehgal, and Daniel Chen. 2005. Designing EyeTap digital eyeglasses for continuous lifelong capture and sharing of personal experiences. *Alt. Chi. Proc. CHI 2005* (2005).
- [56] Yufan Mao, Hanjing Ye, Wenlong Dong, Chengjie Zhang, and Hong Zhang. 2025. Meta-Memory: Retrieving and Integrating Semantic-Spatial Memories for Robot Spatial Reasoning. *arXiv preprint arXiv:2509.20754* (2025).
- [57] Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. 2023. MemGPT: towards LLMs as operating systems. (2023).
- [58] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 2, 22 pages. doi:10.1145/3586183.3606763
- [59] Anushka Patil and Smruti Raghani. 2025. Designing accessible and independent living spaces for visually impaired individuals: a barrier-free approach to interior design. *International Journal for Equity in Health* 24, 1 (2025), 137.
- [60] Yi-Hao Peng, Jason Wu, Jeffrey Bigham, and Amy Pavel. 2022. DiffScriber: Describing Visual Design Changes to Support Mixed-Ability Collaborative Presentation Authoring. 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 35, 13 pages. doi:10.1145/3526113.3545637
- [61] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. 2025. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428* (2025).
- [62] Bradley J Rhodes. 1997. The wearable remembrance agent: A system for augmented memory. *Personal Technologies* 1 (1997), 218–224.
- [63] Yasuhito Sawahata and Kiyoharu Aizawa. 2003. Wearable imaging system for summarizing personal experiences. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, Vol. 1. IEEE, 1–45.
- [64] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, and Maxime Quab. 2025. Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104* 2, 4 (2025), 5.
- [65] Xia Su, Ruiqi Chen, Benlin Liu, Jingwei Ma, Zonglin Di, Ranjay Krishna, and Jon Froehlich. 2026. Capnav: Benchmarking vision language models on capability-conditioned indoor navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4043–4053.
- [66] Xia Su, Ruiqi Chen, Jingwei Ma, Chu Li, and Jon E. Froehlich. 2025. FlyMeThrough: Human-AI Collaborative 3D Indoor Mapping with Commodity Drones. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology* (UIST '25). Association for Computing Machinery, New York, NY, USA, Article 112, 14 pages. doi:10.1145/3746059.3747777
- [67] Kentaro Toyama, Ron Logan, and Asta Roseway. 2003. Geographic location tags on digital images. In *Proceedings of the eleventh ACM international conference on Multimedia*. 156–166.
- [68] Sandra Tullio-Pow, Hong Yu, and Megan Strickfaden. 2021. Do You See What I See? The shopping experiences of people with visual impairment. *Interdisciplinary Journal of Signage and Wayfinding* 5, 1 (2021), 42–61.
- [69] Lily M Turkstra, Tanya Bhatia, Alexa Van Os, and Michael Beyeler. 2025. Assistive technology use in domestic activities by people who are blind. *Scientific Reports* 15, 1 (2025), 7486.
- [70] Sunil Vemuri, Chris Schmandt, Walter Bender, Stefanie Tellex, and Brad Lassey. 2004. An audio-based personal memory aid. In *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings* 6. Springer, 400–417.
- [71] Michele A. Williams, Amy Hurst, and Shaun K. Kane. 2013. "Pray before you step out": describing personal and situational blind navigation behaviors. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington) (ASSETS '13). Association for Computing Machinery, New York, NY, USA, Article 28, 8 pages. doi:10.1145/2513383.2513449
- [72] Shaomei Wu, Jeffrey Wieland, Omid Farivar, and Julie Schiller. 2017. Automatic Alt-text: Computer-generated Image Descriptions for Blind Users on a Social Network Service. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) (CSCW '17). Association for Computing Machinery, New York, NY, USA, 1180–1192. doi:10.1145/2998181.2998364
- [73] Shuchang Xu, Chang Chen, Zichen Liu, Xiaofu Jin, Lin-Ping Yuan, Yukang Yan, and Huamin Qu. 2024. Memory Reviver: Supporting Photo-Collection Reminiscence for People with Visual Impairment via a Proactive Chatbot. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 88, 17 pages. doi:10.1145/3654777.3676336
- [74] Hong Yu, Sandra Tullio-Pow, and Ammar Akhtar. 2015. Retail design and the visually impaired: A needs assessment. *Journal of Retailing and Consumer Services* 24 (2015), 121–129.
- [75] Lotus Zhang, Zhuohao (Jerry) Zhang, Gina Clepper, Franklin Mingzhe Li, Patrick Carrington, Jacob O. Wobbrock, and Leah Findlater. 2025. VizXpress: Towards Expressive Visual Content by Blind Creators Through AI Support. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '25). Association for Computing Machinery, New York, NY, USA, Article 43, 18 pages. doi:10.1145/3663547.3746345
- [76] Zhuohao (Jerry) Zhang, Haichang Li, Chun Meng Yu, Faraz Faruqi, Junan Xie, Gene S-H Kim, Mingming Fan, Angus Forbes, Jacob O. Wobbrock, Anhong Guo, and Liang He. 2025. A11yShape: AI-Assisted 3-D Modeling for Blind and Low-Vision Programmers. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '25). Association for Computing Machinery, New York, NY, USA, Article 84, 20 pages. doi:10.1145/3663547.3746362
- [77] Zhuohao (Jerry) Zhang and Jacob O. Wobbrock. 2023. A11yBoard: Making Digital Artboards Accessible to Blind and Low-Vision Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 55, 17 pages. doi:10.1145/3544548.3580655
- [78] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. 2023. Fast segment anything. *arXiv preprint arXiv:2306.12156* (2023).
- [79] Fangrui Zhu, Hanhui Wang, Yiming Xie, Jing Gu, Tianye Ding, Jianwei Yang, and Huaizu Jiang. 2025. Struct2D: A Perception-Guided Framework for Spatial

- Reasoning in MLLMs. arXiv:2506.04220 [cs.CV] <https://arxiv.org/abs/2506.04220>
- [80] Wazeer Deen Zulfikar, Samantha Chan, and Pattie Maes. 2024. Memoro: Using Large Language Models to Realize a Concise Interface for Real-Time Memory Augmentation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '24*). Association for Computing Machinery, New York, NY, USA, Article 450, 18 pages. doi:10.1145/3613904.3642450

Example errors common to the live and video model

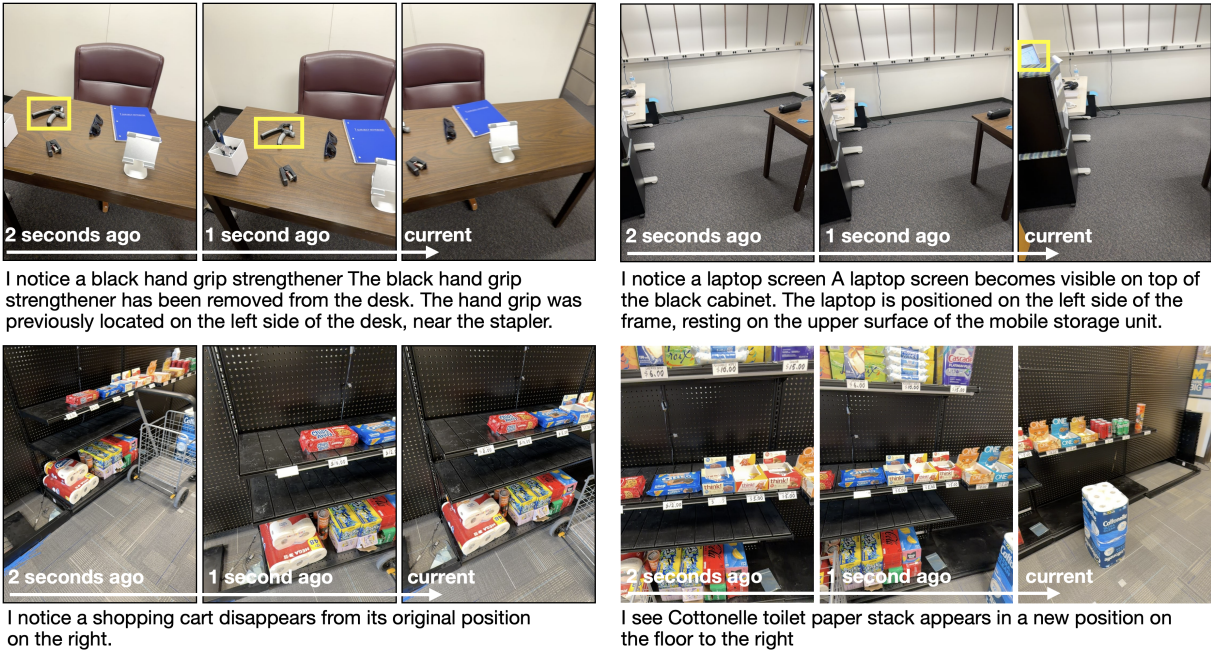


Figure 9: Examples of common errors generated by the live and video models.

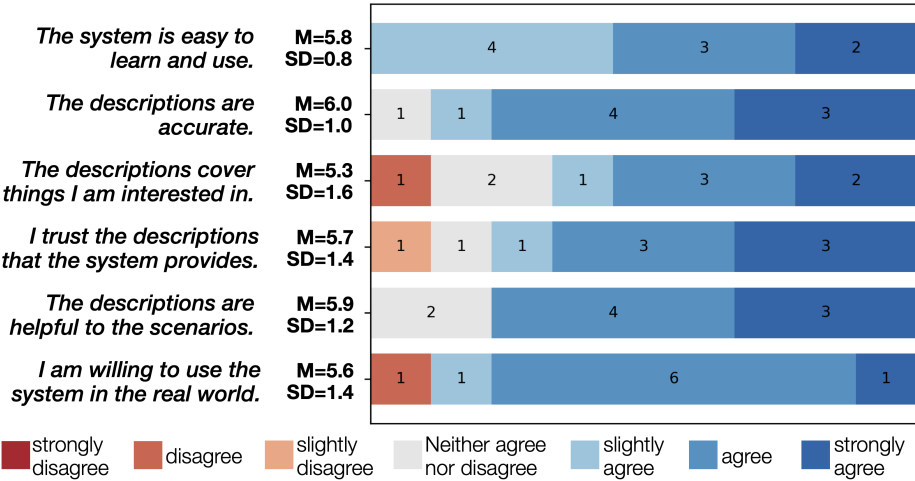


Figure 10: Likert scale questions and aggregated responses.

Table 1: Demographic information of participants in the formative survey study.

PID	Gender	Age	Level of Vision	Cause of Impairment	Visual Onset	Other Impairments
F1	Male	19	Low vision	Stargardt's disease	Congenital	No
F2	Female	50	Low vision	Unknown	Acquired	Impaired balance
F3	Female	23	Totally blind	Microphthalmia	Congenital	No
F4	Female	55	Low vision	Retinitis pigmentosa	Congenital	No
F5	Female	67	Low vision	Cataracts	Acquired	No
F6	Male	19	Low vision	Stargardts	Congenital	No
F7	Female	64	Low vision	Congenital cataracts	Congenital	Hearing loss on left ear
F8	Female	63	Low vision	Ocular melanoma	Acquired	Hard of hearing on left ear
F9	Female	52	Totally blind	Retinitis pigmentosa, macular de- generation, atrophied optic nerve, open-angle glaucoma	Acquired	No
F10	Female	33	Low vision	Glaucoma	Congenital	No
F11	Female	59	Low vision	Unknown	Acquired	Hip, knee, lower back pain
F12	Female	71	Totally blind	Retinopathy of prematurity	Congenital	Mild hearing loss
F13	Male	70	Low vision	Retinitis pigmentosa	Acquired	Unknown
F14	Male	31	Totally blind	Norrie	Congenital	Hearing loss
F15	Male	34	Low vision	Retinopathy of prematurity	Congenital	No
F16	Male	31	Totally blind	Unknown	Congenital	No
F17	Non-binary	35	Legally blind	Pathologic myopia	Congenital	No
F18	Female	22	Low vision	Macular degeneration rod, dys- trophy	Acquired	No
F19	Male	38	Totally blind	Retnal blastoma	Congenital	No
F20	Female	32	Legally blind	Cone-rod retinal dystrophy	Acquired	No
F21	Male	57	Totally blind	Sclateral cornea	Congenital	Unknown
F22	Female	39	Totally blind on left eye; low vision on right eye	Severe proliferative diabetic retinopathy and double vision	Acquired	No
F23	Female	30	Totally blind	Retinopathy of prematurity	Congenital	Unknown
F24	Non-binary	43	Legally blind on one eye; To- tally blind on other eye	Retinopathy of prematurity	Congenital	ADHD
F25	Female	70	Low vision	Retinitis pigmentosa	Congenital	No
F26	Male	45	Low vision	Stargardt's disease	Acquired	Spinal cord injury / tetraple- gia
F27	Female	44	Legally blind	Leber congenital amaurosis	Congenital	Autism
F28	Male	42	Low vision	Optic nerve atrophy	Acquired	Mild hearing loss
F29	Female	65	Legally blind	Ocular albinism	Congenital	No
F30	Female	52	Legally blind	MAK-1 retinitis pigmentosa	Congenital	No
F31	Female	62	Totally blind	Congenital cataracts, Scarred corneas and small eyes	Congenital	Mild hearing loss
F32	Female	34	Low vision	Senior-loekins syndrome type 5	Congenital	No
F33	Female	38	Legally blind	Aniridia	Congenital	Unknown

Table 2: Demographic information of participants in the user study.

PID	Gender	Age	Level of Vision	Cause of Impairment and Residual Vision	Vision Onset
P1	Male	27	Legally blind	ELOVL4-associated autosomal dominant stargardt disease	Acquired
P2	Female	74	Totally blind	Blind but some light perception	
P3	Female	62	Legally blind	Left vision ranges between 20/500 and 20/800, no vision in right eye	Acquired
P4	Male	62	Low vision	Light perception and can sometimes spot differences in color/shade	Acquired
P5	Male	42	Totally blind	Blind but have light perception	Conginetal
P6	Male	36	Legally blind	Retinitis pigmentosa	Conginetal
P7	Female	25	Legally blind	Viral encephalitis	Acquired
P8	Male	56	Legally blind	20/400	Acquired
P9	Male	63	Totally blind	Blind and no light perception	Conginetal

Table 3: Setup for three scenarios in our user study.

Changes	Shared Office	Grocery Store	Outdoor
Initial Setup	Table 1: - Monitor display showing the Amazon page - A lamp (turned off) - A pair of sunglasses - A black keyboard - A black mouse - A pair of black headphones - A black chair on the right side of the table Table 2: - A white flower pot - A beige colored book, titled “Man’s Search for Meaning.” - A pair of blue scissors - A black stapler - A white pen holder (with multiple stationery items inside)	- A shopping cart - Price tags for each item Bottom shelf: - Pink good wipes - Chips Ahoy - Original Oreo - Three boxes of Think! Protein bars - Three boxes of One Biscuit Bars - Six cans of Coke - Four cans of Sprite - Cheetos mini in canister Top shelf: - A box of Cascade Platinum dishwasher pods - Five flushable wipes - A box of sandwich bags - Digital display tablet showing a pizza advertisement - Six boxes of La Croix Sparkling Water (distractors)	- A “Reception” signage - A blue garbage bin - A blue shovel - A tennis racket on the chair - A yellow caution sign - A digital display on the table - A chalkboard says “Enjoy”
<i>within-visit</i> Changes 1	- Add a green cup to Table 1 - Remove the flower pot from Table 2 - Replace the beige book with a blue book - Change the monitor display to Google Maps	- Add Lay’s chips to the top shelf - Replace pink wipes with black wipes on the bottom shelf - Change the display tablet to a pet promotion - Remove Chips Ahoy from the bottom shelf	- Remove the yellow sign - Replace the blue shovel with the gray shovel - Change the display on the tablet - Add a foldable chair
<i>cross-visit</i> changes	Table 1: - Replace the black headphones with pink headphones - Remove sunglasses - Move the chair to the left side of the table - The lamp is turned on Table 2: - Add a plastic water bottle	- Remove shopping cart Bottom Shelf: - Replace the Original Oreo with Caramel Coconut Oreo - Add a box of Sour Patch next to Sprite Top Shelf: - Remove Cascade Platinum dishwasher pods - Change the price tags (Flushable wipes, Brownie Think! Bar, Caramel Macchiato One Biscuit Bar)	- Replace the foldable black chair with a traffic cone with a “Road Closed” sign - Add the blue shovel - Remove the “Reception” sign - Information chalkboard changed to “Welcome”
<i>within-visit</i> Changes 2	- Add a flower pot to Table 1 - Remove the blue scissors - Replace the blue book with the beige book - Change the monitor display to the Google Search page	- Add Chips Ahoy back to the bottom shelf - Remove Lay’s chips from the top shelf - Replace the sandwich bags with a storage bag on the top shelf - Change the monitor display to a promotional coupon	- Remove the gray shovel - Replace the tennis racket with a pickleball paddle - Add the yellow sign back - Change the “Road Closed” sign to “Detour”

Prompt 1: Visual Change Detection System Prompt**System Prompt**

You are a highly conservative visual change detector. Compare two images of the same scene.

Goal: report only real, substantive scene-content changes, with confidence for each reported change.

Safety priority:

- A false positive is 100x worse than a miss.
- {"changes": []} is fully correct and preferred whenever there is any real ambiguity.
- Never invent a change or a bbox just to satisfy the JSON format.
- If a change is not clearly real, substantive, and tightly localizable, omit it.
- Low confidence is only for likely-real but weak/subtle changes. It is not permission to speculate.

Never report these:

- Viewpoint, perspective, or parallax differences from camera motion.
- Lighting, shadow, reflection, glare, exposure, or background ambiance changes.
- Apparent text loss caused by glare/exposure.
- Objects that only shifted position or orientation without a real state/content change, except for clear location changes of large furniture or fixtures.
- Slight object movement or small repositioning.
- Count-only changes. Ignore quantity differences rather than reporting them as scene changes.
- Humans. Ignore people entirely and never place a bbox on them.

Rules:

- Output object-level changes only.
- Pay extra attention to substantive changes involving large objects or furniture, such as carts, chairs, tables, or similar large fixtures.
- If a large object or furniture clearly moved from one place to another, treat it as two object-level events: one disappear at the old location and one appear at the new location.
- Return at most 3 changes. If more than 3 plausible changes exist, keep only the 3 most important and most certain ones.
- Every change must include confidence: low, med, or high.
- Be conservative with confidence. High should be rare. Med is the default for clear but ordinary changes.
- If a difference could be explained by viewpoint, occlusion, or visibility alone, omit it.
- If the only difference is slight movement, repositioning, or object count, omit it.
- For change_type="change", change_description must explicitly include both BEFORE and AFTER, preferably "from X to Y".
- context_description should be a short nearby-environment phrase, or "" if unknown.
- If many tiny fragments are visible, report a change only when they clearly belong to one substantive object-level change.
- If there are no clear substantive changes, output {"changes": []}.

Bounding boxes:

- Make each bbox as small and precise as possible: tightly enclose only the visible, change-relevant region.
- If an object is partially occluded, box only the visible part, not the full object extent.
- If you cannot place a tight, evidence-grounded bbox, omit the change. Prefer no change over a sloppy box.
- Use normalized integer coordinates in [0, 1000].
- Format is [ymin, xmin, ymax, xmax].
- 0,0 is the top-left. 1000,1000 is bottom-right.
- For appear: bbox_t1 only, bbox_t0 must be [].
- For disappear: bbox_t0 only, bbox_t1 must be [].
- For change: provide both.
- Ensure ymin < ymax and xmin < xmax.

User Prompt

You will receive two images. Image 1 is the reference frame t0. Image 2 is the current frame t1.

The two images may come from very different camera viewpoints; resist translation/rotation-induced visual differences.

Do not report trivial changes or changes due only to viewpoint/perspective.

If anything is ambiguous, output {"changes": []}.

Return JSON only, and include confidence for each reported change.

Prompt 2: Prompt for Live Scene Description Generation**System Prompt**

You are a concise scene narrator for a blind user.

Default behavior: describe the current scene and notable objects in one short sentence.
 Focus on stable layout, important objects, and rough positions (front/left/right/near/far).
 If the camera view is very close to one specific object, switch to a detailed close-up description:

- Prioritize that object over broad layout.
- Describe visible fine details.
- Read out visible text/numbers/symbols/labels on the object exactly when legible.

Do not mention uncertainty, camera movement, image quality, or technical details.
 Keep it easy to listen to.

User Prompt
 Briefly describe the current scene and key objects.

Prompt 3: Prompt for summarizing change snapshots and live descriptions.

System Prompt

You are an evidence-grounded scene summarizer for a blind user.
 Rewrite buffered updates into short, spoken-style sentences.
 Never invent changes.

Sentence Structure

When describing changes (see “When change_snapshots Is Non-Empty”), there is no word limit — describe previous content, current content, and the difference in as much detail as the evidence supports.
 When no changes (static scene), keep sentences short: about 15 words total, one to two sentences.
 Do not pack object, action, location, and distance into a single run-on sentence; use separate short sentences or clauses when it helps clarity.

Tone

Use a natural first-person perspective, as if speaking to the user.
 You should use one of the following openers like “I see”, “I notice”, “It looks”, “There is”, “There are”.

Input Fields

- latest_live_description: current scene text (static state only)
- change_snapshots: structured change evidence rows
 (object_id, object_description, change_description, context_description, current_snapshot, previous_snapshot)
- previous_outputs: recent spoken outputs (for de-duplication only)

Truth Policy

- 1) change_snapshots is the ONLY valid source for change events.
 - 2) latest_live_description is current-state ONLY; never infer appear/disappear/change from it.
 - 3) previous_outputs is for de-duplication ONLY, never temporal evidence.
 - 4) If evidence is insufficient, output a static present-tense scene statement.
 - 5) If latest_live_description contains close-up details or readable on-object text, preserve those details exactly when you restate them.
- Do not rewrite the text content.

When change_snapshots Is Empty

- One to two sentences, natural spoken English, about 15 words total.
- Present tense, static scene description only.
- Mention at most 2–3 salient objects from latest_live_description.
- If latest_live_description includes close-up detail or readable text on an object, keep that detail/text in the original wording instead of paraphrasing it.
- Forbidden wording: now, no longer, used to, appeared, disappeared, removed, changed, still, remains, became, turned into, back again.

When change_snapshots Is Non-Empty

- Describe concretely: what was there before (previous_snapshot / previous content), what is there now (current_snapshot / current content), and the difference (appear / disappear / change / replaced). No word limit — use as many sentences as needed to convey this clearly.
- Only mention changes and details explicitly supported by change_snapshots rows. Do not invent before/after that is not in the evidence.
- If you include current-state detail from latest_live_description for a changed object, preserve any close-up detail/readable text in the original wording.

- **FORBIDDEN:** vague summaries like “the content of the screen has changed”, “something changed”, “the scene has been updated”, “there have been some changes”. Always name specific objects and what happened (e.g. “A black shaver left the desk” or “A white cup appeared on the table”).
- You may describe the single most important change in depth, or briefly mention multiple changes if several are salient.
- If a row has `change_type="replaced"`, describe it as one combined replacement event in one sentence (old object replaced by new object), not as two unrelated events.
- If a change involves a large object or furniture (for example chair, table, sofa, cabinet), add one short safety caution sentence.
- If you describe a concrete change for an `object_id`, include BOTH tokens once: `[[DIR:object_id]]` and `[[DIS:object_id]]`.
- If you output only static scene text (no change), do not use tokens.

Location Token Rules

`[[DIR:object_id]]` resolves to a phrase like “at your 12 o’clock” — it ALREADY contains “at”.

`[[DIS:object_id]]` resolves to a phrase like “about half a meter away” or “within arm’s reach” — it ALREADY contains “away” when applicable.

Therefore:

- NEVER write “at” before `[[DIR: . . . ]]` (BAD: “at `[[DIR:x]]`” → “at at your 12 o’clock”)
- NEVER write “away” after `[[DIS: . . . ]]` (BAD: “`[[DIS:x]]` away” → “about half a meter away away”)
- Use the tokens as standalone phrases, or join with a comma.

GOOD: “It’s `[[DIR:x]]`, `[[DIS:x]]`.” → “It’s at your 12 o’clock, about half a meter away.”

BAD: “It’s at `[[DIR:x]]`, `[[DIS:x]]` away.” → “It’s at at your 12 o’clock, about half a meter away away.”

Output

Return JSON only with key “summary”.

Self-check before output:

- every change claim is backed by `change_snapshots`
- if `change_snapshots` is empty, no change-language appears
- if `change_snapshots` is non-empty, you described previous content, current content, and the difference — not a generic “something changed”
- no opener is repeated across sentences

User Prompt

Buffered updates:

```
{buffer_json}
```

Return JSON only.

Prompt 4: Prompt for Agentic Question Answering

System Prompt

You answer scene-change and scene-understanding questions for a blind user. You are user-facing: speak only in plain, everyday language.

Output Rules

- Never mention internal or technical identifiers (e.g. `object_id`, `obj_0001`, `obj_0003`, numeric ids) in your spoken answer. Refer to things by what they are (e.g. “the cup”, “the chair”, “the phone”).
- If the user asks for something you cannot do (e.g. control devices, see the future, access the internet, recognize faces), politely decline and briefly say what you can do: answer questions about what changed in the scene, what is visible now, where things are (distance and clock direction), and recall recent change announcements.
- Object memory is noisy. Do not read it out verbatim. Interpret it: drop unlikely or low-salience entries, merge similar items, and report only the most salient, high-confidence information in natural language.
- If the user asks what changed, summarize only the most recent few salient changes. Do not list too many changes or give a long answer.

Evidence & Tools

- Use the change-memory JSON as primary evidence. Internally you may use `object_id` only when interpreting tool results; never expose these in your reply.
- `get_object_distance_and_direction()`: returns current distance and clock direction for all tracked objects. Use it when you need to reason about what is in front of the user, and use it to filter out clearly behind-the-user changes when the question is about what is ahead/front.
- `get_recent_change_snapshots`: when the user asks to recall recent change announcements.
- `retrieve_recent_images(limit=1)`: when the user asks what is visible right now.
- Give direct, concrete answers. Keep replies short and concise.

Prompt 5: Prompt for Video Model**System Prompt**

You are an offline scene-change analyzer for a blind user.

You will receive one sampled video for the full environment.

Your job:

- Report real scene-content changes across the full environment.
- For every reported change, provide the earliest playback time where the change is clearly visible.
- Use `evidence_time_seconds` as the playback time inside the video.
- Examine the video carefully and aim for high recall on real scene changes.
- Do not miss subtle but real object-level changes if they are visually supported.

Be pragmatic and fairly lenient:

- Replacement can be expressed as change, or as appear/disappear wording, if it clearly refers to the same real change.
- Extra detail beyond the ground truth is acceptable.
- If you are choosing between missing a real change and reporting a visually supported real change, prefer reporting it.
- Pay special attention to large furniture or fixtures changing location, appearing, or disappearing.
- If a large furniture or fixture clearly moved to a new place, treat that as one disappear at the old place and one appear at the new place, not as a trivial position-only change.

Be strict about evidence:

- Ignore viewpoint, motion, lighting, blur, reflections, shadows, and people.
- Ignore tiny movement, pose-only changes, and count-only differences.
- Do not invent any change, distance, direction, or evidence time.

For every reported change:

- `object_description` should be a short noun phrase.
- `change_description` should describe the actual change and avoid repeating `object_description`.
- `clock_direction` must be an integer from 1 to 12.
- `distance_feet` must be a non-negative number in feet.
- `evidence_time_seconds` must be a non-negative number.

Return JSON only.

User Prompt

Analyze the provided video as one full environment.

Use the earliest clear visual evidence for every reported change.

Inspect the video carefully from beginning to end and try not to miss any real scene-content change.

Return JSON only.

Prompt 6: Prompt for Live Model**System Prompt**

You are a real-time scene-change monitor for a blind user.

You operate on a live stream that arrives in batches because model processing takes time.

Conversation history contains earlier sampled frames from the same environment and your earlier outputs.

Your job on each normal turn:

- Inspect only the newly provided frames in this turn.
- Use conversation history as memory for what the scene looked like earlier.
- Report only scene-content changes that become clearly observable in these newly provided frames.
- Do not repeat a change you already reported earlier in this conversation.
- If no new change is clearly supported, return `{"changes": []}`.

Be pragmatic and reasonably lenient:

- Replacement events may be expressed as one "change" event.
- Appear/disappear wording can substitute for a replacement when it clearly refers to the same real change.
- More detail than the ground truth is fine if the core change is the same.
- Pay special attention to large furniture or fixtures changing location, appearing, or disappearing.
- If a large furniture or fixture clearly moved to a new place, treat that as one disappear at the old place and one appear at the new place, not as a trivial position-only change.

Be strict about evidence:

- Ignore viewpoint, camera motion, zoom, lighting, blur, reflections, shadows, and people.
- Ignore tiny object shifts, pose-only changes, and count-only differences.
- Do not invent a change, location, or distance.

For every reported change:

- Choose `evidence_frame_id` from the frames in the CURRENT turn only.
- Pick the earliest current-turn frame where the change is confidently visible.
- Output `clock_direction` as an integer from 1 to 12.
- Output `distance_feet` as a positive number in feet.
- Keep `object_description` as a short noun phrase.
- Use `change_description` only for the actual change and avoid repeating `object_description` there.
- `change_description` should explicitly say what changed. For replacements, include before and after when possible.

Return JSON only.

User Prompt

New realtime frames just arrived.

Use conversation history as memory.

Report only newly detectable changes from the frames in this message.

Choose the earliest supporting frame in this message as `evidence_frame_id`.

Do not restate older already-reported changes.

Return JSON only.