Taylor & Francis
Taylor & Francis Group

International Journal of Human–Computer Interaction

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/hihc20

Context-Adaptive Visual Cues for Safe Navigation in Augmented Reality Using Machine Learning

Arne Seeliger, Raphael P. Weibel & Stefan Feuerriegel

To cite this article: Arne Seeliger, Raphael P. Weibel & Stefan Feuerriegel (2024) Context-Adaptive Visual Cues for Safe Navigation in Augmented Reality Using Machine Learning, International Journal of Human–Computer Interaction, 40:3, 761-781, DOI: 10.1080/10447318.2022.2122114

To link to this article: https://doi.org/10.1080/10447318.2022.2122114

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
2024, VOL. 40, NO. 3, 761–781
https://doi.org/10.1080/10447318.2022.2122114

Taylor & Francis
Taylor & Francis Group

Context-Adaptive Visual Cues for Safe Navigation in Augmented Reality Using Machine Learning

Arne Seeliger*, Raphael P. Weibel*, and Stefan Feuerriegel

ETH Zurich, Zurich, Switzerland

ABSTRACT
Augmented reality (AR) using head-mounted displays (HMDs) is a powerful tool for user navigation. Existing approaches usually display navigational cues that are constantly visible (always-on). This limits real-world application, as visual cues can mask safety-critical objects. To address this challenge, we develop a context-adaptive system for safe navigation in AR using machine learning. Specifically, our system utilizes a neural network, trained to predict when to display visual cues during AR-based navigation. For this, we conducted two user studies. In User Study 1, we recorded training data from an AR HMD. In User Study 2, we compared our context-adaptive system to an always-on system. We find that our context-adaptive system enables task completion speeds on a par with the always-on system, promotes user autonomy, and facilitates safety through reduced visual noise. Overall, participants expressed their preference for our context-adaptive system in an industrial workplace setting.

1. Introduction

By combining the physical and the virtual world, augmented reality (AR) technologies can facilitate effective user navigation. When using head-mounted displays (HMDs), navigation is often achieved by displaying visual cues that guide users toward areas of interest (AOIs). In these cases, the navigational guidance provided is usually identical for all users (Wang et al., 2016). More specifically, visual cues are typically always-on, meaning always visible in the user's field of view (FOV). This is problematic, as it has been shown that AR-based navigation cues can overlap with real objects, thereby leading to distractions and obscured obstacles (Arntz et al., 2020; Krupenia & Sanderson, 2006; Liu et al., 2009). For example, in industrial settings, users need to pay close attention to potentially hazardous elements like heavy machinery or dangerous materials. In these settings, Kim et al. (2016) found that AR HMDs may increase distraction and reduce situational awareness. Therefore, in many situations, an inherent trade-off between effective user navigation and safety arises. This conflict of objectives poses a challenge for the design of AR-based navigation systems.

For AR HMDs, one potential solution to the trade-off between effective user navigation and safety is to give the user control over when to show or hide visual cues. This necessitates explicit user input in the form of hand gestures, gaze, or voice commands. However, explicit input is not only a possible burden to the user (Pfeuffer et al., 2021), but it is, in many situations, difficult to acquire from the user. For example, in healthcare and industrial applications, users often cannot safely re-deploy eye gaze and also rely on the hands-free capability of AR HMDs. Moreover, work environments are frequently subject to noise, rendering voice input unreliable (Arntz et al., 2020). Thus, in many real-world situations, explicit user input is not a solution to the conflict between effective user navigation and safety.

Another remedy regarding the trade-off between effective user navigation and safety is to assess implicit user input and the environment, thus accounting for the user context. In this regard, the notion of *context* comprises of any information which characterizes the situation of a user. AR applications are inherently context-based due to the spatial registration of the AR content alone (Grubert et al., 2017). Going beyond this aspect, user context has been proposed for other challenges in AR. For instance, user context has been used to adapt the content presentation of assistance systems (Katić et al., 2015; Petersen & Stricker, 2015). However, user context has not yet been utilized for balancing the user's need for information with safety during AR-based navigation. Existing research is limited to works that use virtual reality (VR) environments. For example, Alghofaili et al. (2019) adaptively displayed navigational cues during a simulated driving task in VR using a machine learning model based on eye gaze. Burova et al. (2020) also utilized eye gaze to increase safety awareness in an industrial maintenance setting. Their VR system can indicate known areas of risk depending on the user's gaze direction. It is, however, not clear whether findings from VR settings can be directly transferred to AR settings. Regarding AR,

CONTACT Arne Seeliger aseeliger@ethz.ch ETH Zurich, Weinbergstrasse 56/58, Zurich, Switzerland
*These authors contributed equally to this work

research on context-adaptive navigation is limited to recent work by Truong-Allié et al. (2021), who present a system that displays visual guidance based on the detected user activity in multitasking situations. However, this work treats wayfinding and guidance only as auxiliary tasks.

In this article, we develop a context-adaptive system for safe navigation in AR based on machine learning. Specifically, our system adaptively displays visual cues depending on features relating to the current (1) task context (e.g., task progress) and (2) user context (e.g., eye gaze and head movements). To do so, we trained a state-of-the-art neural network to predict when to hide or show visual navigation cues in the user's FOV.

To develop our system and test its effectiveness, we conducted two separate user studies using a simulated industrial task. In User Study 1, we collected data from 15 participants to train a deep neural network. We show that the trained neural network achieves good prediction performance on unseen test data. In User Study 2, we used the trained neural network as part of our context-adaptive system and compared it to a conventional always-on system, which constantly displays visual cues for navigation. This comparison was based on task-related metrics and self-reports from an additional 10 participants who had not participated in User Study 1 and a modified experimental setup. We find that our context-adaptive navigation system based on machine learning allows users to solve the task quickly while promoting user autonomy and benefiting safety through less visual noise. The majority of participants expressed their preference for our context-adaptive system in an industrial work setting.

2. Related work

Our work relates to existing research on (1) visual cues for guidance, specifically navigation, and (2) context in AR.

2.1. Visual cues for guidance and navigation in augmented reality

In AR, navigation toward objects of interest and guidance more generally can be achieved in different ways. In this section, we focus on guidance through visual cues that are displayed by an AR HMD. Such visual cues have been found to effectively guide users, thereby improving task performance metrics, such as completion times (Büttner et al., 2016; Jeffri & Rambli, 2020; Renner & Pfeiffer, 2017; Seeliger et al., 2021).

A variety of visual cues for user guidance have been developed. For instance, Biocca et al. (2006) introduced the well-known omnidirectional attention funnel, which served as inspiration for other representations, such as the augmented tunnel (Hanson et al., 2017; Schwerdtfeger et al., 2011). Moreover, different types of arrows have been studied (Bolton et al., 2015; Gruenefeld, El Ali, et al., 2017; Murauer et al., 2018; Renner & Pfeiffer, 2017, 2020; Schwerdtfeger et al., 2011). For example, *Flying ARrow* was developed to effectively point toward out-of-view objects given small FOVs (Gruenefeld et al., 2018). Related to this are other cues that focus on targets outside the FOV. *Halo* (Baudisch & Rosenholtz, 2003) displays circles around off-screen targets, where the circles reach into the border region of the display window. Similarly, *Wedge* (Gustafson et al., 2008) uses isosceles triangles instead of circles. The spherical wave-based guidance *SWAVE* (Renner & Pfeiffer, 2017) employs waves propagating toward a target. Along the same lines, *EyeSee360* (Gruenefeld, Ennenga, et al., 2017) utilizes radar-like visualizations to achieve guidance toward targets outside the user's FOV. Other cues include the swarm visualization *HiveFive* (Lange et al., 2020) or flickering-based cues (Renner & Pfeiffer, 2017). While some of the aforementioned cues were originally developed for two-dimensional interfaces, such as screens, they have also been found suitable for HMDs (Gruenefeld, El Ali, et al., 2017).

Most of the above visual cues have been implemented for finding objects at short distances and outside the user's FOV. For longer distances, other visual cues have been investigated. Saha et al. (2017) used a path-based cue consisting of large arrows to navigate participants through a supermarket setting in VR. Renner and Pfeiffer (2020) compared the use of a similar path-based cue with other navigational cues to find objects in a complex environment. Arntz et al. (2020) also compared the use of a directional arrow with a path-based navigation cue in an industry setting. The authors found that the path-based cue provided better navigational support but led to larger distances walked by the user. Feedback from the users further showed that the path-based cue proved to be distracting when the route displayed overlapped with the real environment. Our work is directly informed by the research described above. For instance, we utilize a path-based visual cue together with a highlighting box (see Section 3.1 for details).

2.2. Context in AR

Context has been defined as any feature that describes the situation of a user and the environment (Abowd et al., 1999). Schmidt et al. (1999) further distinguish context-related features by splitting them into two different categories: Features concerning human factors, such as the user and task, and features concerning the physical environment, such as the location and surroundings.

Systems using a context to adapt their behavior, so-called context-aware systems, have been applied in many different fields and scenarios (Baldauf et al., 2007; Bettini et al., 2010; Strang & Linnhoff-Popien, 2004). AR HMD devices are usually context-aware by design, as they collect localization information from cameras to detect the position of the device and visualize virtual elements in the real world (Flatt et al., 2015; Grubert et al., 2017). Most of the systems mentioned in Section 2.1 can therefore be seen as context-aware. However, their context does not include features concerning the users and their tasks. In the following, we give an overview of context-aware AR applications for HMDs that include the user's context and use machine learning.

2.2.1. Applications of context-aware AR systems

There are a few systems that combine the localization information of an AR HMD with the context of the user. For example, Lampen et al. (2019) combine the localization information of an AR HMD with a camera-based model detection system that allows to overlay virtual assembly instructions on real manufacturing objects. Saha et al. (2017) simulated an AR HMD with a VR HMD and combined localization information with physiological data to build a context that contains the affective state of the users, which would allow applications to react to a user's change in affect. Katić et al. (2015) used an AR HMD instrument tracking system to assist dental surgeons during implant operations by checking the position and angle of the used instruments and visualizing deviations from the ideal position of the surgeon. Previous research has also incorporated different data, most notably eye gaze. For example, Pfeuffer et al. (2021) combined an AR HMD with an eye-tracker to build a context-aware application, which decides the level of information the user receives about, for example, a conversation partner. Truong-Allié et al. (2021) developed an adaptive guidance system for AR HMDs that uses gaze data in combination with head and hand position to detect a user's current activity. Similarly, albeit using VR to simulate a scenario where AR would be applicable, Alghofaili et al. (2019) used eye-tracking data to implement an adaptive navigation help that showed participants the directions through a virtual city. In this work, we develop a context-adaptive system for navigation similar to Alghofaili et al. (2019). In contrast to the previous work, we present context-adaptive user navigation using actual AR. That is, we perform training, testing, and evaluation in a real environment, instead of a simulated or virtual environment.

2.2.2. Machine learning for AR HMDs

There are many AR systems that use machine learning on mobile devices, such as smartphones (Le et al., 2021; Su et al., 2019). However, there is only a small body of research employing machine learning on AR HMDs. Most of this research focuses on object detection based on the camera information (Knopp et al., 2019; Naritomi & Yanai, 2020). For example, Subakti (2018) implemented a deep-learning image detection module that allows to visualize information on machines in a factory without additional knowledge of the layout of a factory. Similarly, Atzigen et al. (2021) computed the optimal screw placements in a surgical scenario. There are few systems, which use machine learning for other purposes than image detection. For example, David-John et al. (2021) built a gaze-based system that predicts a user's object selection intent. Similarly, Alghofaili et al. (2019) trained a machine learning model using eye-tracking data in their adaptive navigation system. However, both of these systems have been implemented for VR HMDs and use data from one sensor (i.e., eye-tracker) as input for their models. Our system differs from the aforementioned ones, as we build a more general model using data from multiple sensors of an AR HMD. From a technical perspective, our approach is also related to the work of Truong-Allié et al. (2021). Yet, our system detects changing levels of assistance needed during user navigation, rather than detecting different user activities and is based on a different task.

3. Context-adaptive navigation system using machine learning

Our context-adaptive system assists users in navigating indoor environments. Specifically, it visualizes the route between different target objects (referred to as *targets*). It comprises of: (1) A visual cue for navigation, (2) an AR HMD for display, data collection, and computation, and (3) a machine learning model coupled with (4) a decision logic for deciding when to show visual cues.

3.1. Choice of visual cue and navigation route generation

We utilized a directional path (see Figure 1) coupled with a highlighting box (see Figure 2) to navigate users from one target to another. Our choice was based on multiple design

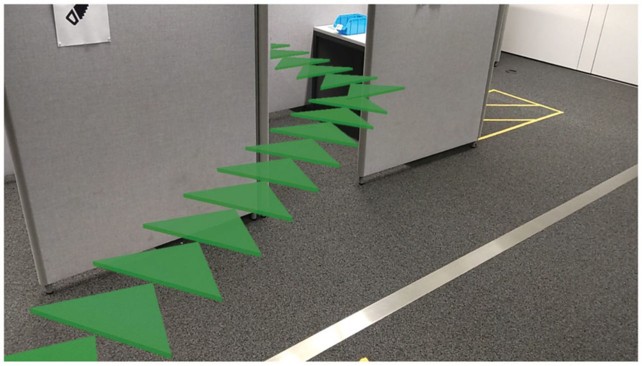


Figure 1. Path to target box when a target is not within the FOV.

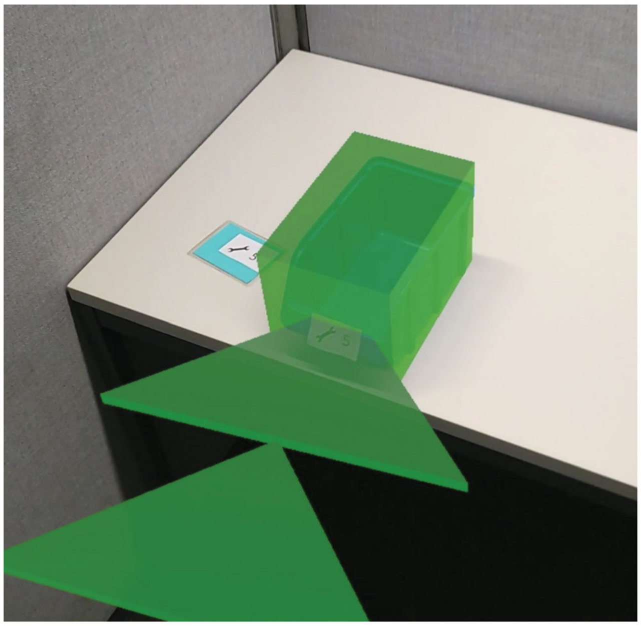


Figure 2. Highlighting box when a target is within the FOV.

Table 1. Context data is extracted at each time step.

Context type	Context data	Description
Task	Task progress	Percentage of target items retrieved.
	Task duration	Seconds since the start of the task.
	Subtask duration	Seconds since the retrieval of the last target item.
User	Eye gaze angle	Visual angle θ_i at time t_i between head direction vector f_i and gaze direction vector g_i. This is calculated through the dot product of two three-dimensional vectors f_i and g_i, i.e., $\theta_i = \arccos\left(\frac{\langle f_i, g_i \rangle}{\langle \lvert f_i \rvert, \lvert g_i \rvert \rangle}\right)$.
	Eye gaze change	Angular difference between gaze direction vectors g_i and g_{i-1}.
	Head movement	Head movement in three-dimensional space is expressed as a rotation matrix $M \in R^{3\times3}$.
	AOI fixated	The AOI currently gazed at, is represented as a one-hot encoded vector $A \in R^5$. A description of the five defined AOIs is given in the text below and details regarding how the AOIs relate to the user studies are provided in Section 4.1.
	Distance to target	Number of meters from the user location to the current target location is measured along the path of the visual cue.
	Depth map	A matrix $D \in R^{20\times20}$ containing the distances in meters from the user's head position to the next object along the head direction vector. We utilize the spatial mapping capacity of the HMD to acquire a set of points in three-dimensional space, which approximate the real-world surroundings.
	Last cue	Seconds since the visual navigation cue was last displayed.

considerations that have been discussed in the literature as reviewed in Section 2.1. First, we utilized a path-based visual cue instead of, for example, floating arrows, as users preferred this type of navigational assistance in prior studies (Arntz et al., 2020). While path-based visual navigation cues have been criticized for covering up a large portion of the user's FOV (Arntz et al., 2020), this downside is mitigated by the fact that our visual cue is only shown for short periods of time.

Second, Renner and Pfeiffer (2020) found that path-based visual cues should be augmented with directional information. Therefore, we opted to augment our path-based visual cue by using directional elements (triangles) pointing in the direction of the next target. To keep the cue inside the FOV of the user while still allowing them to see what's in front, we placed the path 80 centimeters above the ground. Third, when the actual target was in view, we highlighted it by superimposing a semi-transparent box as shown in Figure 2. This kind of visual cue has been shown to work effectively for targets that are within the user's FOV (Seeliger et al., 2021).

Lastly, research indicates that participants prefer visual cues that leave them some autonomy (Renner & Pfeiffer, 2020). Our task differs from the task used in the latter study as we used a predefined order of targets (see Section 4.1). However, we still allowed the user to deviate from the proposed route by constantly recomputing the shortest path from the current user position. We achieved this by employing a grid of (invisible) virtual waypoints throughout the room. Specifically, the waypoints were placed at every intersection of the room. The shortest path was then computed along those waypoints using Dijkstra's algorithm (Dijkstra, 1959). Further implementation details of the visual cue are given in Appendix A.

3.2. AR HMD device and context data

Our system is designed to run on Microsoft's HoloLens 2 in conjunction with Microsoft's Mixed Reality Toolkit (MRTK) and Unity. The device provides a FOV for augmented viewing of 43° horizontally and 29° vertically. Moreover, previous research found that HoloLens 2 offers an eye tracking system with a maximum sampling frequency of 30 Hz and a range of ~40° in both directions horizontally and a vertical range of ~20° in the upper direction and 40° in the lower direction (Seeliger et al., 2021). The latter study further found that the eye tracking system of HoloLens 2 measures accurately (0.51° at 1 m distance) and precisely ($SD = 0.30$ at 1 m distance). The HMD also provides speech as a user input option, which is used especially for data collection (User Study 1).

3.2.1. Context data

From the AR HMD, we recorded different signals relating to (1) the task context and (2) the user context (see Table 1). This data was dynamically computed by accessing multiple AR HMD sensors (i.e., visible light head-tracking cameras, far-depth cameras, inertial measurement unit, and eye gaze tracking cameras) for a given time step $t_i \in T$, resulting in time series. To extract context data related to spatial aspects (e.g., distance to target), we created a virtual model of the surroundings of the experiment using the spatial mapping capabilities of HoloLens 2 and Unity. This virtual model was aligned with the actual room using a printed QR code, which users scanned upon starting the system.[1] We sampled all sensor signals with ~60 Hz, except for the eye gaze data, which could only be sampled at a maximum of 30 Hz. We, therefore, upsampled this data to 60 Hz by using each value twice.

Five AOIs have been defined. These are closely related to the chosen experimental setup of our user studies as described in Section 4.1. (1) *Target* represents the current target object that a user wants to navigate to. In our experimental setup, a target was one of many small boxes that were placed across a room (see Figure 6). (2) *Non-Targets* represent all boxes that are not the current target. Additional AOIs were defined based on the room of our experimental setup, which contained different isles. The boxes placed within each isle shared certain characteristics (i.e., similar content as described in Section 4.1). On the entrance to an isle, a label attached to the isle wall depicted this information similar to a street sign. Hence, we defined

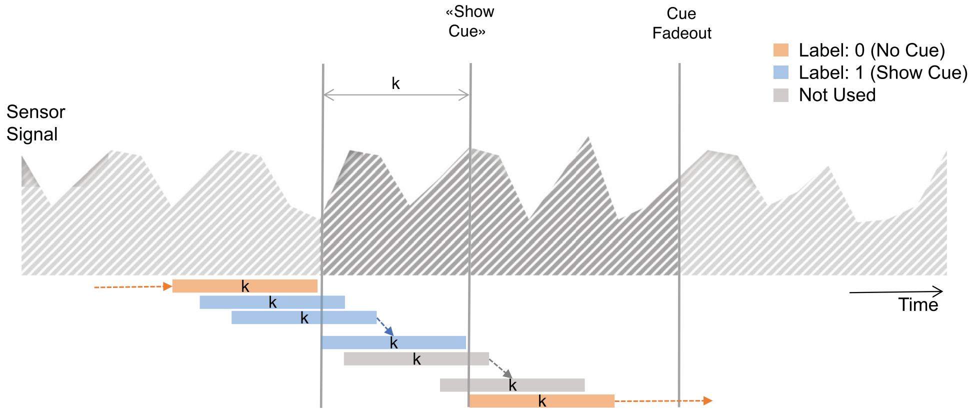


Figure 3. Sliding window for feature generation and labeling.

(3) *Target Isle Label* to describe the label of the isle that contained the target. (4) *Non-Target Isle Labels* represent all other isle labels (i.e., all labels of isles that do not contain the current target). Lastly, (5) *Cue* describes the visual navigation cue. To gather data on AOIs, we computed the intersection of each gaze ray at a time t_i with the aligned virtual model of the spatial surroundings.

Note that the position of the user within the environment was not utilized as context data.[2] This was done to enable the machine learning model to learn from generalizable user behavior rather than certain locations unique to a given (training) environment.

3.2.2. User input data

We further recorded two types of user input data. First, users issued the voice command *next target* whenever they reached the current target. We used this signal to ensure that the navigation cue would always lead to the user's subsequent target. Second, to train the machine learning model, we recorded when users wanted to receive visual navigation. Users could issue the voice command *show cue*, which would show the visual cue for 2.5 s[3] and store the time point of the user input. All voice commands were acoustically confirmed by the HMD via a short ringing sound.

3.3. Machine learning model

We aimed at predicting whether to hide or show the visual navigation cue at a given time step t_i. This outcome was modeled as a binary variable $y_i \in \{0, 1\}$. We utilized features X_i computed from the HMD context and user input data in a predictive function $f(X_i, X_{i-1}, ..., X_{i-k}) = \hat{y}_i$. Here, k denotes the number of time lags that are included as part of the feature vector, as described below.

3.3.1. Feature generation and labeling

We used a sliding window with a window size of k time steps to generate feature vectors together with their associated label from the context and user input data. This is depicted in Figure 3. Here, the colored boxes represent feature vectors, each with a length of k time steps. The respective labels of each feature vector are shown in different colors. We encoded $y_i = 1$ for all feature vectors that end within one window size before when a voice command *show cue* was issued. Similar to Alghofaili et al. (2019), this was done because it can be assumed that participants felt the need for navigational guidance before issuing the voice command. Feature vectors during which the visual cue was triggered were not used for training. All other feature vectors were labeled with $y_i = 0$. The window size k was chosen as part of the hyper-parameter tuning, which is described in Section 3.3.3.

3.3.2. Model architecture

To predict the binary outcome variable y, we utilized a deep neural network consisting of multiple input branches as depicted in Figure 4. Specifically, for the depth map input, we employed a 2D-convolution layer followed by a 2D-pooling layer. The resulting output was flattened and used as input for a fully connected layer. Similarly, for the AOI input as well as the eye gaze and head movement inputs, we used 1D-convolutions and 1D-pooling, followed by a flattening layer and a fully connected layer. The remaining input branches were concatenated and the resulting output was fed into another fully connected layer. All resulting outputs of the respective input branches were subsequently concatenated and used as input for a fully connected layer. We used ReLu activation functions for all the above layers. The last layer was a fully connected layer with one unit and Sigmoid activation.

We utilized a neural network for two reasons. First, prior research (Alghofaili et al., 2019; Truong-Allié et al., 2021) successfully applied this model for related scenarios. Second, neural networks allow for an easy combination of different types of input data (i.e., numerical data and image data). We also tested other neural network architectures. In particular, a fully connected neural network resulted in inferior performance, which is why we used the above model architecture.

3.3.3. Training and hyper-parameter tuning

We trained the neural network by minimizing the binary cross-entropy loss between true and predicted labels using the NAdam optimizer (Dozat, 2016) and a batch size of 256. To address imbalances in the distribution of class labels, we computed balanced class weights as follows. The weight of the minority class was set to $\frac{n}{2\sum_i y_i}$, where n represents the

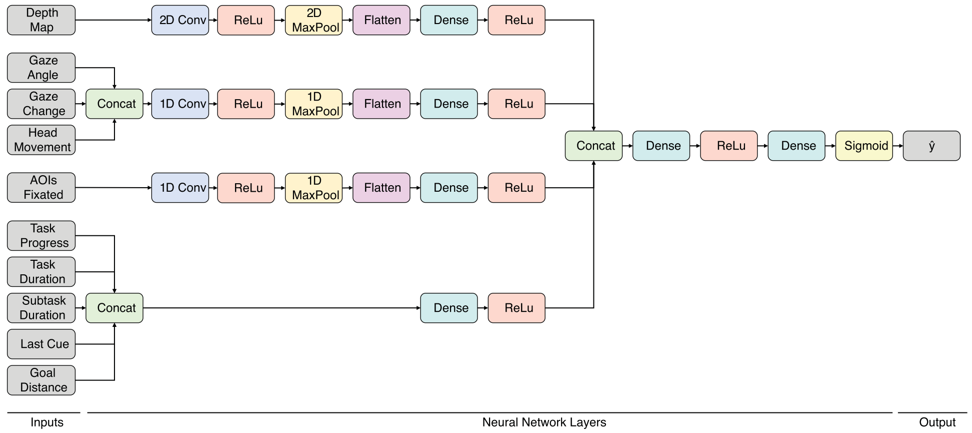


Figure 4. Model architecture which consists of multiple input branches.

number of samples and y_i the binary label ($=1$ if cue was displayed, 0 otherwise).

We split the gathered data into a training (~90%) and testing set (~10%). From the training data, we used around 10% of the data samples as validation data, which was held out during training to optimize different hyper-parameters. These included the number of units in each but the last layer of the neural network, as well as the number of convolutional filters, the kernel sizes for 1D- and 2D-convolutions, the pooling window size, and the learning rate. For hyperparameter tuning, we employed Hyperband (Li et al., 2018), a bandit-based random search approach, which often outperforms state-of-the-art Bayesian optimization methods. Hyperband efficiently allocates a resource budget (e.g., number of training iterations) to different sets of hyperparameters by successively dropping sets that do not perform well. We allocated a budget of 25 training iterations and chose as our performance measure the area under the receiver operating characteristic curve (ROC) of the validation data (i.e., validation AUC). To avoid overfitting, we made use of early stopping with a patience of three. Appendix B lists the search grid used during hyper-parameter tuning and reports the best values found.

After determining the final model architecture, we retrained the model on the entire training data set for 50 epochs using a batch size of 256. The test data was held out, and therefore not used during training. To account for overfitting, we utilized early stopping with a patience of eight epochs. The training was performed on an NVIDIA Tesla V100 GPU. Details regarding the data gathered and the model performance on this data are given in Section 5.1.

Additionally, we trained the model with fewer inputs to investigate its sensitivity to the input features (sensitivity test), as shown in Appendix C. All three variations resulted in lower accuracy and precision. Recall, on the other hand, was slightly higher. We later discuss the implications of using models with potentially fewer inputs in Section 6.

3.4. Displaying visual cues using machine learning model predictions

Our system adaptively displayed visual cues for navigation based on the trained neural network. Specifically, the inference was run once every second and the system displayed the visual cue presented above for 2.5 s if the model output was greater than a threshold value of $J = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} + \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} - 1 \approx 0.45$. This value is known as the Youden Index, and it is frequently used to choose an optimal threshold for binary classifiers (Fluss et al., 2005). If the visual cue was shown, no inference was run to save computational resources.

The system made use of a client-server architecture. More specifically, the machine learning model was deployed on a server, which received data through an HTTP POST request from the AR HMD. The total time between data transfer and return of the prediction to the AR HMD was around 180–220 ms.

4. User studies

To develop and test the effectiveness of the proposed context-adaptive system for safe navigation in AR based on machine learning, we conducted two user studies. In User Study 1, we collected training data by computing features from sensor signals of the AR HMD alongside the points in time when users chose to show or hide visual navigation cues. Subsequently, we trained the neural network to predict when to display visual cues for navigation in the user's FOV. In User Study 2, we utilized the trained neural network in our context-adaptive system and investigated its effectiveness and usability. To investigate model generalizability, we recruited a different set of participants and modified the experimental setup, which is described in more detail in Section 5.2.

4.1. Task design and setup

We conceived a task in which participants were asked to navigate within a room to find and retrieve items from boxes located at different tables. Figure 5 shows a 180° view of the room (and its full layout is displayed in Appendix D). Each table was surrounded by three partitioning walls, thus forming isles of cubicles. We placed two boxes on each table (see Figure 6). We chose this task and set up as it is representative of many search-and-pick tasks, which are common

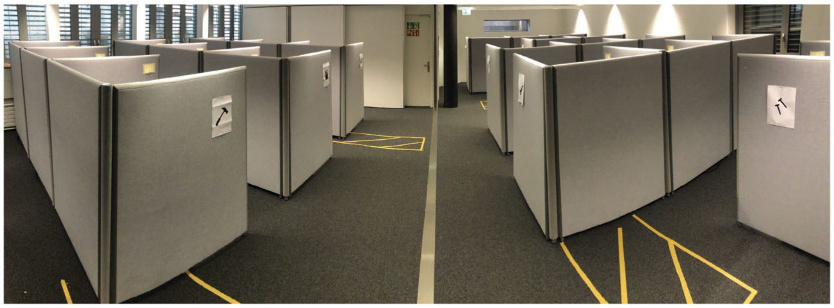


Figure 5. Experimental setup (180° view) with zones marked as hazardous (yellow).

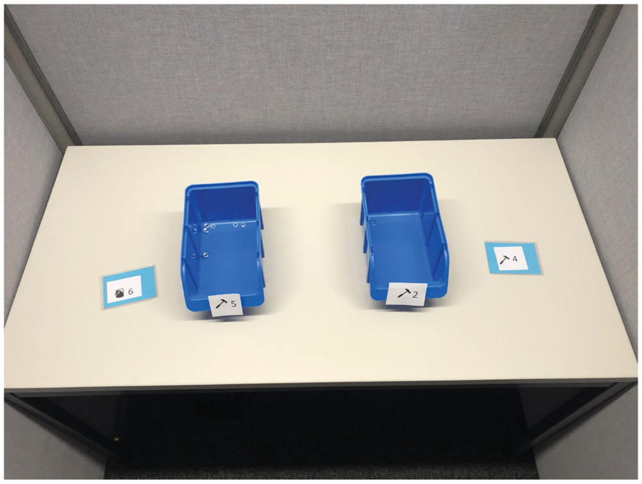


Figure 6. Table with two labeled boxes surrounded by three partitioning walls.

in the industry (e.g., order picking) (Guo et al., 2015). Through our setup, we also follow recent calls to conduct user studies in larger environments where occlusions have to be accounted for (Renner & Pfeiffer, 2020).

The goal of the task was to repeatedly retrieve an item from one of the boxes and bring it to another box. Each box was uniquely identifiable through labels: Every isle of cubicles was labeled with a symbol (similar to a street sign) and every box was labeled with a symbol and number, which resulted in one unique code per box. Within each isle of cubicles, the numbers on the boxes were distributed randomly, so that participants could not rely on an ordering or sorting scheme among the isles. Next to each box, we placed a small card containing the label of the next target box as depicted in Figure 6. We used a separate table as the starting location, where participants received a small card stating the label of the first target box. When participants reached a target box, they were instructed to place the card they were holding at that time into the box and pick up the card laying next to that box to determine the label of the subsequent target box.

During the entire task, each participant wore the AR HMD on which our navigation system was running. In User Study 1, participants could ask the system for navigational guidance in the form of the visual cue, using the voice command *show cue*. This would show the cue for 2.5 s. In User Study 2, the same visual cue was shown to a different set of participants, this time, however, automatically without explicit user input as described in Section 3.

The room layout also included zones marked as hazardous. Participants were instructed to avoid these zones by not stepping into them. We used orange tape to mark the zones, as shown in Figure 5. The motivation behind this was to encourage spatial attention of the participants, thereby preventing an ongoing use (spamming) of the navigational cue. In other words, spamming would lead to many occlusions of the zones marked as hazardous and, therefore, make it hard to avoid them. We further elaborate on this in the context of User Study 1 and User Study 2 in Sections 5.1.1 and 5.2.1, respectively. This design also increased the level of realism in the simulated industry setting.

For both User Study 1 and User Study 2, participants were told that there was no time constraint to the task but that they should finish as quickly as possible while trying not to step into any of the zones marked as hazardous.

5. Evaluation

User Study 1 served as a means to collect data for training the machine learning model. We describe this study and report performance metrics of the trained machine learning model (i.e., AUC score) in Section 5.1. In User Study 2, the trained machine learning model was used to adaptively show visual cues during user navigation. We provide details on this study in Section 5.2 and report task-related measures (e.g., task completion speeds) as well as user self-reports (e.g., perceived workload and usability).

Before conducting both user studies, we obtained ethics approval from the Ethics Committee of ETH Zurich (EK 2021-N-25). For each of the two studies, participants received an information sheet summarizing the goals, methods, and compensation of the user study. Participants were further informed that their participation was voluntary. Participation was not permitted for participants with binocular vision disorder, such as strabismus (eye misalignment, crossed, or wandering eye). After having time to ask any remaining questions, participants signed a consent form and filled out a demographics questionnaire. Participants were compensated with the equivalent of USD 20.

5.1. User Study 1: Data collection for training the machine learning model

In User Study 1, we collected data to train and evaluate the performance of our neural network.

5.1.1. Procedure

Participants conducted the experimental task described in Section 4.1 as follows. Each participant first completed a training round involving eight targets, followed by two blocks of 15 targets each, with a short break in between. The order of the blocks was determined randomly. In the training round, participants were familiarized with both voice commands and the goal of the task. Also, participants were immediately informed if they stepped into a zone marked as hazardous during this time (but not after the training round). In between the two blocks, one experimenter changed the location of three of the zones marked as hazardous, while the respective participant waited in a dedicated waiting room. Appendix D shows the specific locations of these zones. This was done to prevent learning effects for the participants, meaning a memorization of the zones marked as hazardous, which would increase the likelihood of constant use of the navigational cue (spamming, see Section 4.1). Therefore, this helped in ensuring that the neural network learned based on user behavior data, rather than specific environment locations. A change of location of hazardous zones is also likely in industrial settings, thereby increasing the generalizability of the collected data.[4]

For User Study 1, we recruited 15 participants (9 male, 6 female) aged between 22 and 34 ($M = 27.4$; $SD = 3.16$). No participant suffered from color vision impairment or other (vision) disorders. Further, 13 participants had normal vision and two had corrected vision. None of the participants had experience working in industrial settings.

5.1.2. Collected data

We collected $n = 204\,200$ samples across all participants and blocks. Each block had an average duration of 164 s ($SD = 41$) and participants spent, on average, 11.8 s ($SD = 3.5$) on each target (see Figure 7). Using the participants' time per target and the shortest possible path between two targets,[5] we computed participants' task completion speed, which was 1.062 m/s ($SD = 0.3$ m/s) on average (see Appendix F). Each participant issued the voice command *show cue* on average 8.1 times ($SD = 6.1$) per block and, on average, 0.57 times ($SD = 0.65$) per target. Furthermore, we found that the participant's use of the voice command had a large spread with some participants using it up to 1.5 times per target, whereas others did not use it for every target (see Figure 8).

5.1.3. Performance of machine learning model

Here, we report the performance of the trained neural network on the test data, which was held out during training. The confusion matrix at a threshold of $J \approx 0.45$ is given in Figure 9. The neural network achieved an accuracy of 0.71, a recall of 0.85, and a precision of 0.37. Furthermore, to provide a threshold-independent measure of performance, Figure 10 depicts the ROC curve of the trained neural network. The corresponding AUC amounts to 0.81.

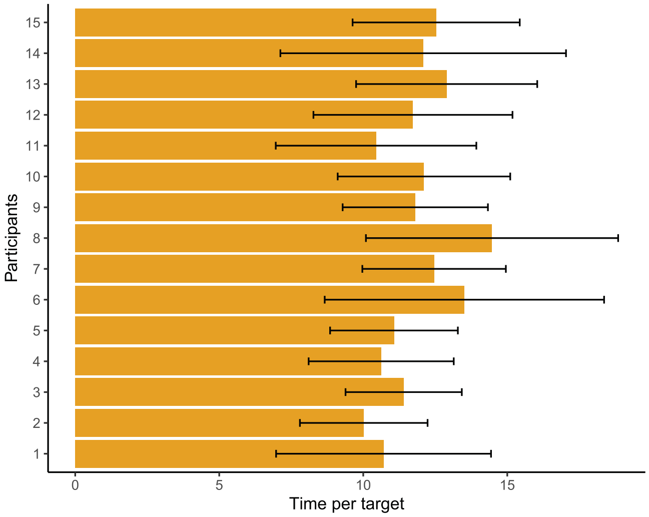


Figure 7. Mean time per target in seconds for each participant in User Study 1. Whiskers show standard deviations.

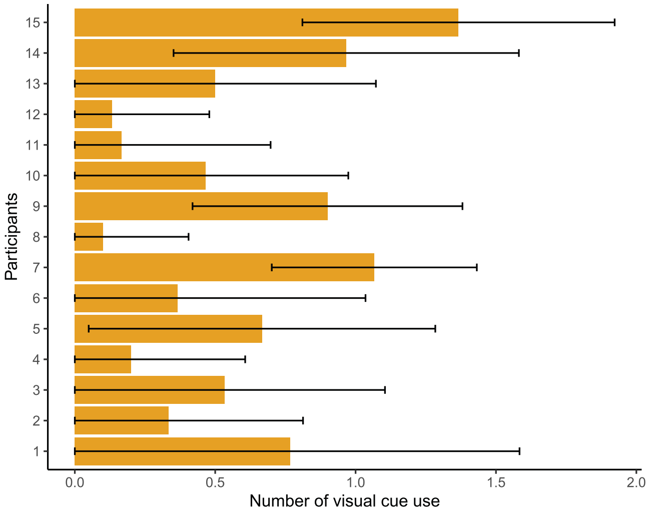


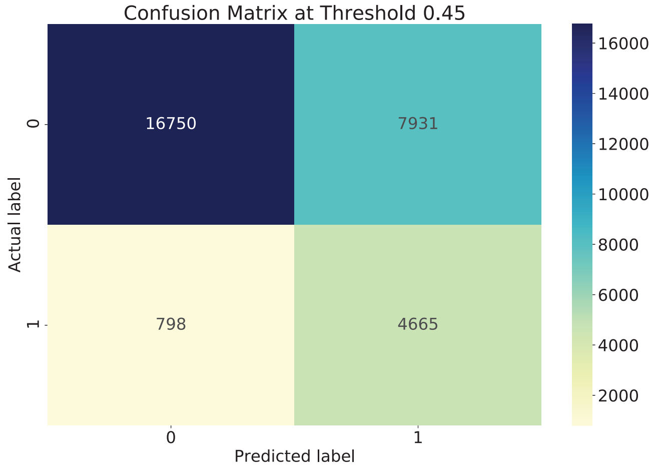
Figure 8. Mean number of activations of the visual cue for each participant per target in User Study 1. Whiskers show standard deviations.

5.2. User Study 2: Objective and subjective evaluation of context-adaptive navigation in AR

The goal of User Study 2 was to compare our context-adaptive system based on machine learning with a common always-on system. To investigate generalizability, we recruited different participants as compared to User Study 1 and modified the experimental setup as described below.

5.2.1. Procedure

As before, participants conducted two experimental blocks with 15 targets each. However, for one of the two blocks, users were supported by our context-adaptive system. For the other block, users were shown the same visual cue, but in an always-on manner. Participants were randomly assigned to start with either of the two blocks. The training round was also slightly adapted. Instead of having one training round consisting of eight targets before the two



Accuracy	Recall	Precision
0.7104	0.8539	0.3704

Figure 9. Confusion matrix at a threshold of 0.45 and values of accuracy, recall, and precision.

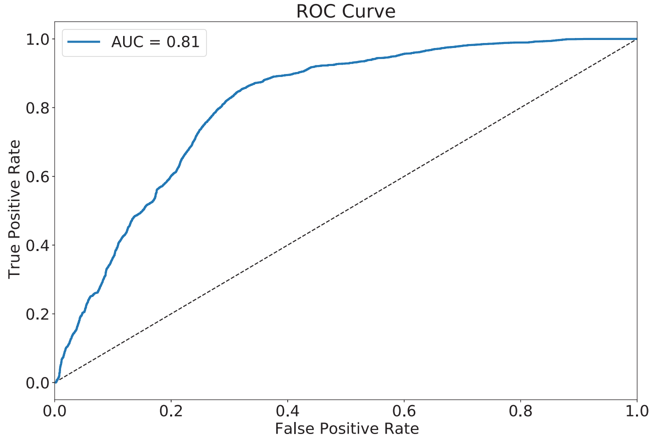


Figure 10. ROC curve of trained neural network.

experimental blocks, participants could familiarize themselves with each experimental condition in one smaller training round consisting of four targets right before the start of the respective experimental block. This way, participants were exposed to the second experimental condition only after completing the first. Moreover, we altered the locations of the zones marked as hazardous (see Appendix D) and the locations of the target boxes compared to User Study 1. Although the machine learning model did not use the location of the user or the zones marked as hazardous as features, we applied these changes to ensure that training and testing conditions differed, thereby assessing the generalizability of the model. We also changed the location of the zones marked as hazardous between the blocks of User Study 2 with the same rationale as in User Study 1, to prevent learning effects for the participants and to increase the realism of the experimental task.

For User Study 2, we recruited 10 participants (three male, seven female) aged between 24 and 41 ($M = 28.22$; $SD = 4.50$). None of them had taken part in User Study 1. Further, no participant suffered from color vision impairment or other (vision) disorders. Eight participants had normal vision and two participants had corrected vision. No participant had experience working in industrial settings.

5.2.2. Task-related metrics and self-reports

We collected both task-related metrics and participants' self-reports as part of User Study 2. Regarding task-related metrics, we recorded the completion time for each individual target. We used this measure together with the shortest possible path between subsequent target boxes to calculate the average speed of completing the task (as described in Section 5.1.2). To investigate workload, we used the six subscales of the NASA TLX (raw) questionnaire (Hart & Staveland, 1988). To assess the usability of the system, we employed the System Usability Scale (SUS) (Bangor et al., 2008). Moreover, we were interested in whether the visual cues were perceived as useful or disturbing. Hence, we additionally asked participants to answer the following questions regarding their user experience on a 5-point Likert scale:[6]

1. I felt distracted by the visual cue. (Distraction)
2. I had to search for the targets a lot. (Search for targets)
3. The visual cue was there when I needed it. (Visible when needed)
4. The visual cue should have been visible more often. (Show more)
5. The visual cue should not have been visible so often. (Show less)
6. The visual cue was useful. (Useful)

We further asked the participant to choose which type of visual cue they would prefer in a workplace environment, such as a factory or warehouse. Finally, we asked the participants to answer the following open questions:

1. What did you like or dislike about the adaptive visual cue?
2. How did the adaptive visual cue help or disturb you?

We analyzed the qualitative data (i.e., the responses of the participants to the open questions) with the goal to identify patterns or themes. After reading the transcripts, two researchers coded the responses iteratively to create a list of codes. We then generated themes from the codes collaboratively using an inductive approach. We chose verbatim quotations (see Section 5.2.3) to highlight the themes relevant to our research objectives.

5.2.3. Results

The average task completion speed of our context-adaptive system was on a par with that of the always-on system. The mean task completion speed was 1.049 m/s ($SD = 0.21$ m/s) for our context-adaptive system and 1.055 m/s ($SD = 0.22$ m/s) for the always-on system. In the context-adaptive system, the visual cue was shown to participants on average 2.4 times ($SD = 1.4$) per target (see Figure 11).

Participants also rated the usability of the systems similarly. Our context-adaptive system achieved a mean of 76 (*SD* = 14.4) on the SUS, and the always-on system a mean of 78 (*SD* = 13.7). The results of the NASA TLX (raw) show that both systems were rated low with respect to the workload, as depicted in Figure 12, and with no statistically significant difference (see Appendix H). Conversely, the overall performance for both systems was rated very highly. As the number of times the visual cue was shown could influence the results of the NASA TLX, we performed a correlation analysis for each of the NASA TLX dimensions. However, we did not find a significant correlation (see Appendix I).

We found that participants perceived both systems as highly useful based on the additional user experience questions asked (as shown in Figure 13). Overall, the user experience was comparable between both systems. For example, participants perceived the frequency of both visual cues as neither too high nor too low. While both types of visual cues received positive feedback, we observed a larger spread within the answers for the context-adaptive cue. Furthermore, the user experience questionnaire results indicate that participants had to search for targets slightly more using the context-adaptive system. In this context, the analysis of the open-ended questions revealed that three of the 10 participants were confused by the inner workings of the context-adaptive system. For example, one participant stated: "*It was a bit confusing, I did not know if it was working or not.*" This confusion, however, was the only negative aspect of the context-adaptive system that was mentioned by participants.

Overall, eight of the ten participants preferred our context-adaptive system over a cue always-on system in a work environment, such as a factory or warehouse. In this context, the open-ended questions yielded additional feedback. We found that participants preferred the context-adaptive system for two main reasons. First, it gave them more autonomy (four participants), as expressed by one participant: "*I actually had to think and check more, made me feel more competent.*" Second, the context-adaptive system resulted in less visual noise (two participants). In this context, it was stated by one participant that "*when I got to know the environment a bit more it was nice to relax the eyes a bit.*" Moreover, two participants specifically acknowledged the assistance through the shown routes. For example, one participant stated that "[the visual cue] *lead me to the right place if I did not know where to go.*"

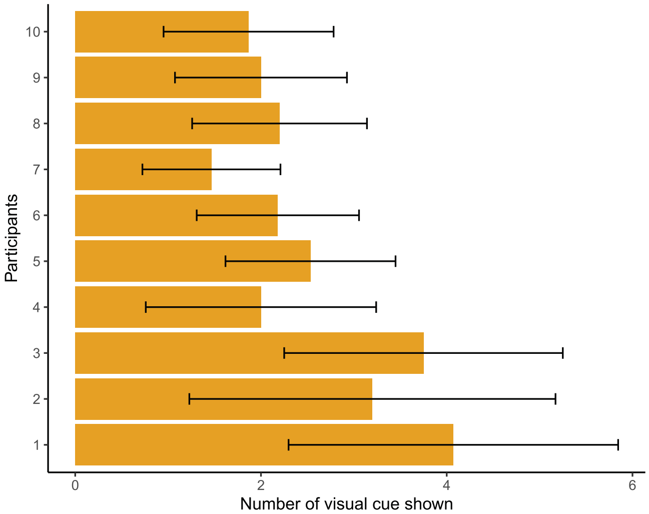


Figure 11. Mean number of activations of the visual cue per participant in User Study 2. Whiskers show standard deviations.

6. Discussion

In this work, we developed a context-adaptive system for user navigation in AR based on machine learning. In the

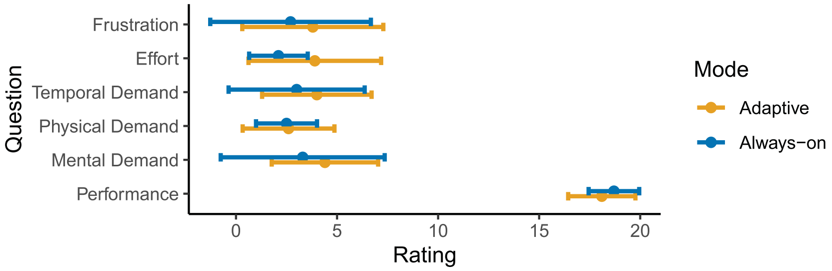


Figure 12. Mean responses to NASA TLX (raw). Whiskers show standard deviations.

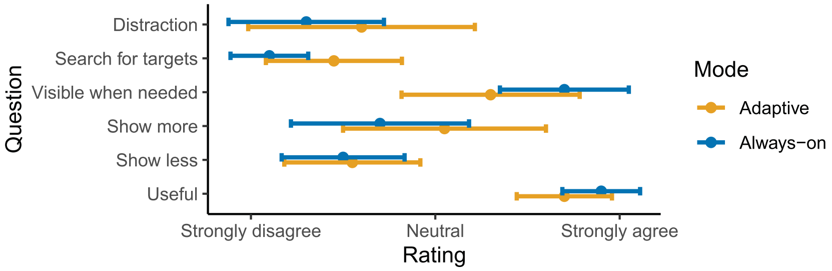


Figure 13. Mean responses to additional user experience questions. Whiskers show standard deviations.

following, we first discuss our work from a technical perspective, followed by a user-centered discussion.

6.1. Technical discussion

From a technical point of view, our context-adaptive system relies on two main features: the visual cue and the machine learning model. Regarding the chosen visual cue, the results from User Study 1 and User Study 2 indicate that the path-based visual cue worked well in the given experimental setting. This corroborates existing research (Arntz et al., 2020) highlighting the effectiveness of path-based visual cues.

Regarding the trained machine learning model, we observed good performance on held-out training data (i.e., the accuracy of 0.71, recall of 0.85). However, the neural network only achieved a precision of 0.37, which indicates room for performance improvements. A direct comparison of these results with prior work is challenging due to differences in used algorithms, tasks, and environments. For instance, Truong-Allié et al. (2021) report an accuracy of 0.86 for their activity recognition model, in which guidance is only an auxiliary task. Further, Alghofaili et al. (2019) report accuracy, precision, and recall of over 0.90 for their VR-based machine learning model given test data on a known virtual environment and accuracy of around 0.80 given an unknown virtual environment. Their system, however, is not only based on VR but also geared toward eye gaze input and a different neural network architecture. Overall, the prediction performance of the machine learning component in our AR system is thus of similar magnitude as that of other works from the VR domain and AR systems for other tasks (activity recognition as opposed to navigation).

Some participants stated that they did not understand the inner workings of the context-adaptive visual cue. Based on this user feedback, we additionally performed an inquiry into the importance of our model input features (see Appendix E), which also addresses the call for assessing the interpretability of machine learning models, especially when considering safety-critical systems (Doshi-Velez & Kim, 2017). Fixated AOIs and the distance to the target had a strong impact on the model prediction, while the time since the visual cue was displayed last and features relating to eye gaze had a low impact. By providing these additional model insights, we hope to promote acceptance of our context-adaptive system for potential future application in industrial and safety-critical settings.

We expect our overall approach to context-adaptive user navigation in AR to generalize to other indoor environments as only the depth map feature is depending directly on the used environment. For this, two considerations should be taken into account. First, our system as a whole requires spatial knowledge of the environment (e.g., virtual model of the room) to compute data, such as the distance to a target. Such data often exists for industrial workplaces or it can be acquired with relatively low effort using the spatial mapping feature of AR HMDs like HoloLens 2. Second, it is likely that our system can be improved further through additional re-training or fine-tuning. Our overall approach can also be applied to other forms of AR-based user guidance, such as guided assembly (Seeliger et al., 2022) as we trained our machine learning model on features that have been computed from standard sensor signals of a consumer-grade HMD. In this context, it is also worth highlighting that it is possible to reduce the computational cost of the system by utilizing fewer input features as part of the machine learning model, thereby sacrificing some level of predictive performance, depending on the feature. For example, omitting the depth map might be a viable alternative for a resource-scarce setting in which all computation is performed exclusively on a wearable device.

6.2. User-centered discussion

When assessing the usability and user experience, our results suggest that the context-adaptive visual cue provided navigational guidance that was similar in effectiveness to a traditional always-on cue. Specifically, completion speeds were almost identical between the two. Hence, our results indicate that efficient navigation in AR does not exclusively rely on visual cues that are always in the user's FOV. Instead, efficient navigation can also be achieved via context-adaptive visual cues, which increase the safety of navigating real workplaces. Our work, therefore, presents a meaningful step toward resolving the conflict between effective user navigation and safety.

These findings extend previous work (Kim et al., 2019), which found that job performance declined when AR-based visual cues are shown only for a limited amount of time after a request by the user, as compared to an always-on system. We, therefore, show that using context-adaptive visual cues, which increase worker safety, does not necessarily come at the cost of job performance. Specifically, user feedback revealed that visual noise was reduced, thereby corroborating existing research (Kim et al., 2019). This is an important aspect in real-world applications of AR HMDs, as prior studies emphasized that information presented in AR HMDs must not reduce awareness of safety hazards (Kim et al., 2016). However, AR applications can be prone to excessively drawing attention to virtual content, an effect known as attention tunneling (Syiem et al., 2021). Using a context-adaptive system, such as the one presented in this work, can provide the means for user navigation that is both effective and safe, especially in workplaces, such as factories or warehouses. Users also indicated a greater level of autonomy. This is in line with work by Renner and Pfeiffer (2020), who found that users prefer guidance techniques that leave some autonomy to them.

Overall, both the context-adaptive and the always-on system received similar, positive workload and usability assessments from the users. This, too, indicates that context-adaptive systems can successfully bridge the gap between user-friendly navigation and safety. Still, more work in this field needs to be conducted to advance the user experience of context-adaptive AR systems for navigation even further. In our study, for example, users had to search for the targets slightly more often using the context-adaptive system as compared to the always-on system. Multiple factors might be the reason for this observation. On one hand, it is possible that the timing of the visual cues was not ideal, leading to an increased mental effort and distraction for the user. From the large spread of responses regarding the frequency of the context-adaptive visual cue, we presume that context-adaptive user navigation in AR also depends on individual user preferences. This is further corroborated by the results of User Study 1, which revealed that the participants' need for the visual cue also exhibited a large spread. Thus, for real-world applications, users might benefit from having the option to control the sensitivity or threshold of the machine learning model, similar to Alghofaili et al. (2019). On the other hand, the system was novel to the users and not every user understood its inner workings sufficiently, which could have led to distractions, too.

Eight out of 10 participants expressed their preference for using the context-adaptive system in an industrial workplace setting. Again, different factors might have contributed to this result. Specifically, users might have valued the inherent safety benefits of the context-adaptive system more than the marginal differences in usability. The preference for the context-adaptive system found in this study contrasts with recent work by Truong-Allié et al. (2021), in which users did not have a clear preference for either adaptive or always-on visual cues. Yet other research on AR HMDs found that users prefer visual cues that are always on (Kim et al., 2019). The latter, however, did not account for potential safety hazards and used simple visual cues that were not context-adaptive. In our study, some observations also favor an always-on system. The always-on system resulted in less mental effort, presumably because it requires less attention shifting due to its constant visibility. Moreover, the predictability and timing of the context-adaptive visual cue might be optimized even more. The choice of user guidance systems thus strongly depends on the task and the user requirements. Overall, more research is needed to investigate the question of user preference regarding adaptive and always-on cues in industrial workplace settings.

The benefits of our context-adaptive system in a real workplace setting also depend on its prediction performance. As stated above, our system achieved a relatively low precision, which indicates a high number of false positive classifications. This means the visual cue appeared more often than needed, thereby obstructing the user's FOV. Since such misclassifications could theoretically happen at any moment in time, the visual cue might appear in potentially dangerous situations. Therefore, the implications for user safety in case of misclassification (or system failure) are an important aspect to consider. One potential remedy could be a manual override, allowing users to hide or show the visual cue in cases of misclassifications. This could also have a beneficial effect on users as participants stated their preferences for autonomy. Additionally, tracking the amount of manual override would further allow to refine a model during retraining.

6.3. Limitations and future research

This work is subject to several limitations that provide avenues for additional research. First, although the number of participants taking part in our user studies is in line with previous research (Alghofaili et al., 2019; David-John et al., 2021; Katić et al., 2015; Lampen et al., 2020; Saha et al., 2017; Truong-Allié et al., 2021), the machine learning model could benefit from additional training data. More participants could also contribute to making more generalizable conclusions regarding the usability and user experience of our system.

Second, both user studies were run in a laboratory, simulating a real-world work environment. Given the novelty of our system and for ensuring participant safety, we opted for this controlled environment. However, this setup might not have triggered the same level of caution as a real-world work environment with actual safety hazards. Additionally, even though the size of the laboratory was relatively large, its dimensions and complexity lag behind a real factory or warehouse environment. This might have allowed participants to learn the layout quickly and remove their need for assistance. However, visual inspection of the progression of the task completion speed over time (in User Study 1 and 2) and the use of visual cues (in User Study 1) did not reveal a pattern indicative of learning (see Appendix G). In this context, it is also important to highlight that the participants did not have industry experience and their behavior might differ from professional workers who perform tasks like order picking regularly. We, therefore, plan to apply our system in the field, for example in a manufacturing setting. Our system could also be transferred to other settings where it is unfeasible to acquire direct user input, for instance, retail stores (Cruz et al., 2019) and supermarkets (Saha et al., 2017).

Finally, our results suggest that the context-adaptive system can display visual cues more often than needed, which has implications for user safety. Therefore, future work should strive to apply safety verification techniques to context-adaptive systems, which are based on machine learning. We also plan to provide the option to manipulate the model threshold so that users themselves can determine an acceptable trade-off between false negative and false positive classifications.

7. Conclusion

In this article, we developed a system for context-adaptive user navigation in AR using machine learning. We collected data from an HMD and trained a neural network to predict when to hide or show navigational visual cues during a picking task (User Study 1). We evaluated our context-adaptive system through a second user study (User Study 2). We found that context-adaptive user navigation can be of great benefit in industrial environments because it enables task completion speeds on a par with always-on navigation while promoting user autonomy and safety through reduced visual noise. The work presented in this article, therefore, provides a meaningful step toward safe, yet efficient, navigation in

AR, which can also be transferred to other forms of AR-based user guidance.

Notes

1. During the user studies (see Section 4), we verified the precision of the alignment between virtual model and actual room for each participant visually, by temporarily showing transparent boxes overlaying the physical elements in the room (e.g., boxes and isle labels). This was done by observing the front-camera feed of HoloLens 2 on a separate computer. The alignment error was estimated to be between one and five centimeters.
2. Other signals extracted also do not allow to infer the user position. For instance, the depth map signal describes the proximity of the user to larger physical objects, which means that different user locations can lead to similar values for the depth map. Likewise, the fixated AOIs only describe which areas of interest have been gazed at by a user, which can happen from a variety of positions within the environment.
3. This duration was selected based on preliminary tests with three users.
4. Even though participants were not informed of these changes, only one participant stepped into a zone marked as hazardous.
5. That is, not necessarily the path that participant walked.
6. We used a 5-point Likert scale, to achieve comparability to Alghofaili et al. (2019) and Renner and Pfeiffer (2020). Question abbreviations as used in results paragraphs are given in brackets.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Arne Seeliger http://orcid.org/0000-0001-7127-8104
Raphael P. Weibel http://orcid.org/0000-0002-8854-7507
Stefan Feuerriegel http://orcid.org/0000-0001-7856-8729

References

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., Steggles, P. (1999). Towards a Better Understanding of Context and Context-Awareness. In H. W. Gellersen, (eds), *Handheld and Ubiquitous Computing. HUC 1999: Vol. 1707. Lecture Notes in Computer Science* (pp. 304–307). Springer. https://doi.org/10.1007/3-540-48157-5_29

Alghofaili, R., Sawahata, Y., Huang, H., Wang, H. C., Shiratori, T., & Yu, L. F. (2019). Lost in style: Gaze-driven adaptive aid for VR navigation. In *Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3290605.3300578

Arntz, A. et al. (2020). Navigating a Heavy Industry Environment Using Augmented Reality-A Comparison of Two Indoor Navigation Designs. In J. Y. C. Chen, & G. Fragomeni, (eds), *Virtual, Augmented and Mixed Reality. Industrial and Everyday Life Applications. HCII 2020: Vol 12191. Lecture Notes in Computer Science* (pp. 3–18). Springer. https://doi.org/10.1007/978-3-030-49698-2_1

Atzigen, M. v., Liebmann, F., Hoch, A., Bauer, D. E., Snedeker, J. G., Farshad, M., & Fürnstahl, P. (2021). HoloYolo: A proof of concept study for marker less surgical navigation of spinal rod implants with augmented reality and on device machine learning. *The International Journal of Medical Robotics + Computer Assisted Surgery*, *17*(1), 1–10. https://doi.org/10.1002/rcs.2184

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(4), 263. https://doi.org/10.1504/IJAHUC.2007.014070

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, *24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Baudisch, P., & Rosenholtz, R. (2003). Halo: A technique for visualizing off-screen objects. In *Conference on Human Factors in Computing Systems* (pp. 481–488). https://doi.org/10.1145/642611.642695

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, *6*(2), 161–180. https://doi.org/10.1016/j.pmcj.2009.06.002

Biocca, F., Tang, A., Owen, C., & Xiao, F. (2006). Attention funnel: Omnidirectional 3D cursor for mobile augmented reality platforms. In *Conference on Human Factors in Computing Systems* (pp. 1115–1122). https://doi.org/10.1145/1124772.1124939

Bolton, A., Burnett, G., & Large, D. R. (2015). An investigation of augmented reality presentations of landmark-based navigation using a head-up display. In *Conference on Automotive User Interfaces and Interactive Vehicular Applications* (pp. 56–63). https://doi.org/10.1145/2799250.2799253

Burova, A., Mäkelä, J., Hakulinen, J., Keskinen, T., Heinonen, H., Siltanen, S., & Turunen, M. (2020). Utilizing VR and gaze tracking to develop AR solutions for industrial maintenance. In *Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376405

Büttner, S., Funk, M., Sand, O., & Röcker, C. (2016). Using head-mounted displays and *in-situ* projection for assistive systems: A comparison. In *Conference on Pervasive Technologies Related to Assistive Environments*. https://doi.org/10.1145/2910674.2910679

Cruz, E., Orts-Escolano, S., Gomez-Donoso, F., Rizo, C., Rangel, J. C., Mora, H., & Cazorla, M. (2019). An augmented reality application for improving shopping experience in large retail stores. *Virtual Reality*, *23*(3), 281–291. https://doi.org/10.1007/s10055-018-0338-3

David-John, B., Peacock, C., Zhang, T., Murdison, T. S., Benko, H., & Jonker, T. R. (2021). Towards gaze-based prediction of the intent to interact in virtual reality. In *Eye Tracking Research and Applications Symposium (ETRA)*. https://doi.org/10.1145/3448018.3458008

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. https://doi.org/10.1007/BF01386390

Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning* (No. 1702.08608v2). http://arxiv.org/abs/1702.08608

Dozat, T. (2016). Incorporating Nesterov momentum into Adam. In *ICLR Workshop*.

Flatt, H., Koch, N., Rocker, C., Gunter, A., & Jasperneite, J. (2015). A context-aware assistance system for maintenance applications in smart factories based on augmented reality and indoor localization. In *Conference on Emerging Technologies & Factory Automation (ETFA)*. https://doi.org/10.1109/ETFA.2015.7301586

Fluss, R., Faraggi, D., & Reiser, B. (2005). Estimation of the Youden index and its associated cutoff point. *Biometrical Journal. Biometrische Zeitschrift*, *47*(4), 458–472. https://doi.org/10.1002/bimj.200410135

Grubert, J., Langlotz, T., Zollmann, S., & Regenbrecht, H. (2017). Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, *23*(6), 1706–1724. https://doi.org/10.1109/TVCG.2016.2543720

Gruenefeld, U., El Ali, A., Heuten, W., & Boll, S. (2017). Visualizing out-of-view objects in head-mounted augmented reality. In *Conference on Human–Computer Interaction with Mobile Devices and Services* (pp. 1–7). https://doi.org/10.1145/3098279.3122124

Gruenefeld, U., Ennenga, D., Ali, A. E., Heuten, W., & Boll, S. (2017). EyeSee360: Designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Symposium on Spatial User Interaction* (pp. 109–118). https://doi.org/10.1145/3131277.3132175

Gruenefeld, U., Lange, D., Hammer, L., Boll, S., & Heuten, W. (2018). FlyingARrow: Pointing towards out-of-view objects on augmented reality devices. In *International Symposium on Pervasive Displays* (Vol. 18, pp. 1–6). https://doi.org/10.1145/3205873.3205881

Guo, A., Wu, X., Shen, Z., Starner, T., Baumann, H., & Gilliland, S. (2015). Order picking with head-up displays. *Computer Magazine*. *48*(6), 16–24. https://doi.org/10.1109/MC.2015.166

Gustafson, S., Baudisch, P., Gutwin, C., & Irani, P. (2008). Wedge: Clutter-free visualization of off-screen locations. In *Conference on Human Factors in Computing Systems* (pp. 787–796). https://doi.org/10.1145/1357054.1357179

Hanson, R., Falkenström, W., & Miettinen, M. (2017). Augmented reality as a means of conveying picking information in kit preparation for mixed-model assembly. *Computers & Industrial Engineering*, *113*(April), 570–575. https://doi.org/10.1016/j.cie.2017.09.048

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, *52*(C), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Jeffri, N. F. S., & Rambli, D. R. A. (2020). Guidelines for the interface design of AR systems for manual assembly. In *Proceedings of the 2020 International Conference on Virtual and Augmented Reality Simulations* (pp. 70–77). https://doi.org/10.1145/3385378.3385389

Katić, D., Spengler, P., Bodenstedt, S., Castrillon-Oberndorfer, G., Seeberger, R., Hoffmann, J., Dillmann, R., & Speidel, S. (2015). A system for context-aware intraoperative augmented reality in dental implant surgery. *International Journal of Computer Assisted Radiology and Surgery*, *10*(1), 101–108. https://doi.org/10.1007/s11548-014-1005-0

Kim, S., Nussbaum, M. A., & Gabbard, J. L. (2016). Augmented reality "smart glasses" in the workplace: Industry perspectives and challenges for worker safety and health. *IIE Transactions on Occupational Ergonomics and Human Factors*, *4*(4), 253–258. https://doi.org/10.1080/21577323.2016.1214635

Kim, S., Nussbaum, M. A., & Gabbard, J. L. (2019). Influences of augmented reality head-worn display type and user interface design on performance and usability in simulated warehouse order picking. *Applied Ergonomics*, *74*(1), 186–193. https://doi.org/10.1016/j.apergo.2018.08.026

Knopp, S., Klimant, P., Schaffrath, R., Voigt, E., Fritzsche, R., & Allmacher, C. (2019). HoloLens AR – Using Vuforia-based marker tracking together with text recognition in an assembly scenario. In *International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (pp. 63–64). https://doi.org/10.1109/ISMAR-Adjunct.2019.00030

Krupenia, S., & Sanderson, P. M. (2006). Does a head-mounted display worsen inattentional blindness? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(16), 1638–1642. https://doi.org/10.1177/154193120605001626

Lampen, E., Lehwald, J., & Pfeiffer, T. (2020). A context-aware assistance framework for implicit interaction with an augmented human. In *HCII 2020: Virtual, Augmented and Mixed Reality. Industrial and Everyday Life Applications* (pp. 91–110). https://doi.org/10.1007/978-3-030-49698-2_7

Lampen, E., Teuber, J., Gaisbauer, F., Bär, T., Pfeiffer, T., & Wachsmuth, S. (2019). Combining simulation and augmented reality methods for enhanced worker assistance in manual assembly. *Procedia CIRP*, *81*(3), 588–593. https://doi.org/10.1016/j.procir.2019.03.160

Lange, D., Stratmann, T. C., Gruenefeld, U., & Boll, S. (2020). HiveFive: Immersion preserving attention guidance in virtual reality. In *Conference on Human Factors in Computing Systems* (pp. 1–13). Virtual Conference. https://doi.org/10.1145/3313831.3376803

Le, H., Nguyen, M., Yan, W. Q., & Nguyen, H. (2021). Augmented reality and machine learning incorporation using YOLOv3 and ARKit. *Applied Sciences*, *11*(13), 6006. https://doi.org/10.3390/app11136006

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, *18*(1), 1–52. https://doi.org/10.48550/arXiv.1603.06560

Liu, D., Jenkins, S. A., Sanderson, P. M., Watson, M. O., Leane, T., Kruys, A., & Russell, W. J. (2009). Monitoring with head-mounted displays: Performance and safety in a full-scale simulator and part-task trainer. *Anesthesia & Analgesia*, *109*(4), 1135–1146. https://doi.org/10.1213/ANE.0b013e3181b5a200

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 4768–4777).

Murauer, N., Schön, D., Müller, F., Pflanz, N., Günther, S., & Funk, M. (2018). An analysis of language impact on augmented reality order picking training. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference* (pp. 351–357). https://doi.org/10.1145/3197768.3201570

Naritomi, S., & Yanai, K. (2020). CalorieCaptorGlass: Food calorie estimation based on actual size using HoloLens and deep learning. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (pp. 818–819). https://doi.org/10.1109/VRW50115.2020.00260

Petersen, N., & Stricker, D. (2015). Cognitive augmented reality. *Computers and Graphics*, *53*(8), 82–91. https://doi.org/10.1016/j.cag.2015.08.009

Pfeuffer, K., Abdrabou, Y., Esteves, A., Rivu, R., Abdelrahman, Y., Meitner, S., Saadi, A., & Alt, F. (2021). ARtention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics*, *95*(2), 1–12. https://doi.org/10.1016/j.cag.2021.01.001

Renner, P., & Pfeiffer, T. (2017). Attention guiding techniques using peripheral vision and eye tracking for feedback in augmented-reality-based assistance systems. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 186–194). https://doi.org/10.1109/3DUI.2017.7893338

Renner, P., & Pfeiffer, T. (2020). AR-glasses-based attention guiding for complex environments. In *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments* (pp. 1–10). https://doi.org/10.1145/3389189.3389198

Saha, D. P., Knapp, R. B., & Martin, T. L. (2017). Affective feedback in a virtual reality based intelligent supermarket. In *Adjunct Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers* (pp. 646–653). https://doi.org/10.1145/3123024.3124426

Schmidt, A., Beigl, M., & Gellersen, H.-W. (1999). There is more to context than location. *Computers & Graphics*, *23*(6), 893–901. https://doi.org/10.1016/S0097-8493(99)00120-X

Schwerdtfeger, B., Reif, R., Günthner, W. A., & Klinker, G. (2011). Pick-by-vision: There is something to pick at the end of the augmented tunnel. *Virtual Reality*, *15*(2–3), 213–223. https://doi.org/10.1007/s10055-011-0187-9

Seeliger, A., Merz, G., Holz, C., & Feuerriegel, S. (2021). Exploring the effect of visual cues on eye gaze during AR-guided picking and assembly tasks. In *International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (pp. 159–164). https://doi.org/10.1109/ISMAR-Adjunct54149.2021.00041

Seeliger, A., Netland, T., & Feuerriegel, S. (2022). Augmented reality for machine setups: Task performance and usability evaluation in a field test. *Procedia CIRP*, *107*(3), 570–575. https://doi.org/10.1016/j.procir.2022.05.027

Strang, T., & Linnhoff-Popien, C. (2004). A context modeling survey. In *Proceedings of the Workshop on Advanced Context Modeling, Reasoning and Management*.

Su, Y., Rambach, J., Minaskan, N., Lesur, P., Pagani, A., & Stricker, D. (2019). Deep multi-state object pose estimation for augmented reality assembly. In *Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)* (pp. 222–227). https://doi.org/10.1109/ISMAR-Adjunct.2019.00-42

Subakti, H. (2018). Indoor augmented reality using deep learning for Industry 4.0 smart factories. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (pp. 63–68). https://doi.org/10.1109/COMPSAC.2018.10204

Syiem, B. V., Kelly, R. M., Goncalves, J., Velloso, E., & Dingler, T. (2021). Impact of task on attentional tunneling in handheld augmented reality. In *Conference on Human Factors in Computing Systems* (pp. 1–14). https://doi.org/10.1145/3411764.3445580

Truong-Allié, C., Paljic, A., Roux, A., & Herbeth, M. (2021). User behavior adaptive AR guidance for wayfinding and tasks completion. *Multimodal Technologies and Interaction*, *5*(11), 65–81. https://doi.org/10.3390/mti5110065

Wang, X., Ong, S. K., & Nee, A. Y.-C. (2016). Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, *30*(3), 406–421. https://doi.org/10.1016/j.aei.2016.05.004

About the authors

Arne Seeliger is a researcher and PhD candidate at ETH Zurich. He investigates the combination of human intelligence with algorithmic intelligence and enjoys conducting this empirical research in the field. Arne holds a bachelor's degree in Technology and Management and a master's degree in Business Analytics and Computer Science.

Raphael P. Weibel is a researcher and PhD candidate at the Mobiliar Lab for Analytics at ETH Zurich. He investigates the use of Mixed Reality and wearable technologies to facilitate user understanding of visualized data in different scenarios. He holds a bachelor's and a master's degree in Computer Science.

Stefan Feuerriegel is a full professor at LMU Munich School of Management where he heads the Institute of AI in Management. Previously, he was an assistant professor at ETH Zurich. In his research, Stefan develops, implements, and evaluates Artificial Intelligence technologies that improve management decision-making.

Appendix A. Implementation details for visual cue

The visual cue was visualized along the shortest path between the current location of the user and the target. The shortest path was computed along a set of waypoints using Dijkstra's algorithm (Dijkstra, 1959). The waypoints were placed at every intersection of the room (i.e., there were waypoints in front of all cubicles and in the empty space between the cubicles). The cue consisted of green (RGBA: 0, 255, 20, 178) triangles lined up on the shortest path pointing toward the next waypoint on the path or the target if the target was closer than any of the waypoints. Each triangle was 22 cm long, 41 cm wide, and 1.5 cm high. The number of triangles on each segment (i.e., between waypoints, between the last waypoint and the target, or between the user and the first waypoint) of the path was computed based on the length of a segment. Specifically, the distance to the next waypoint was divided by the length of one triangle. When approaching a waypoint, the triangle closest to the user was removed once the distance required a smaller number of triangles.

Appendix B. Search grid for hyper-parameters

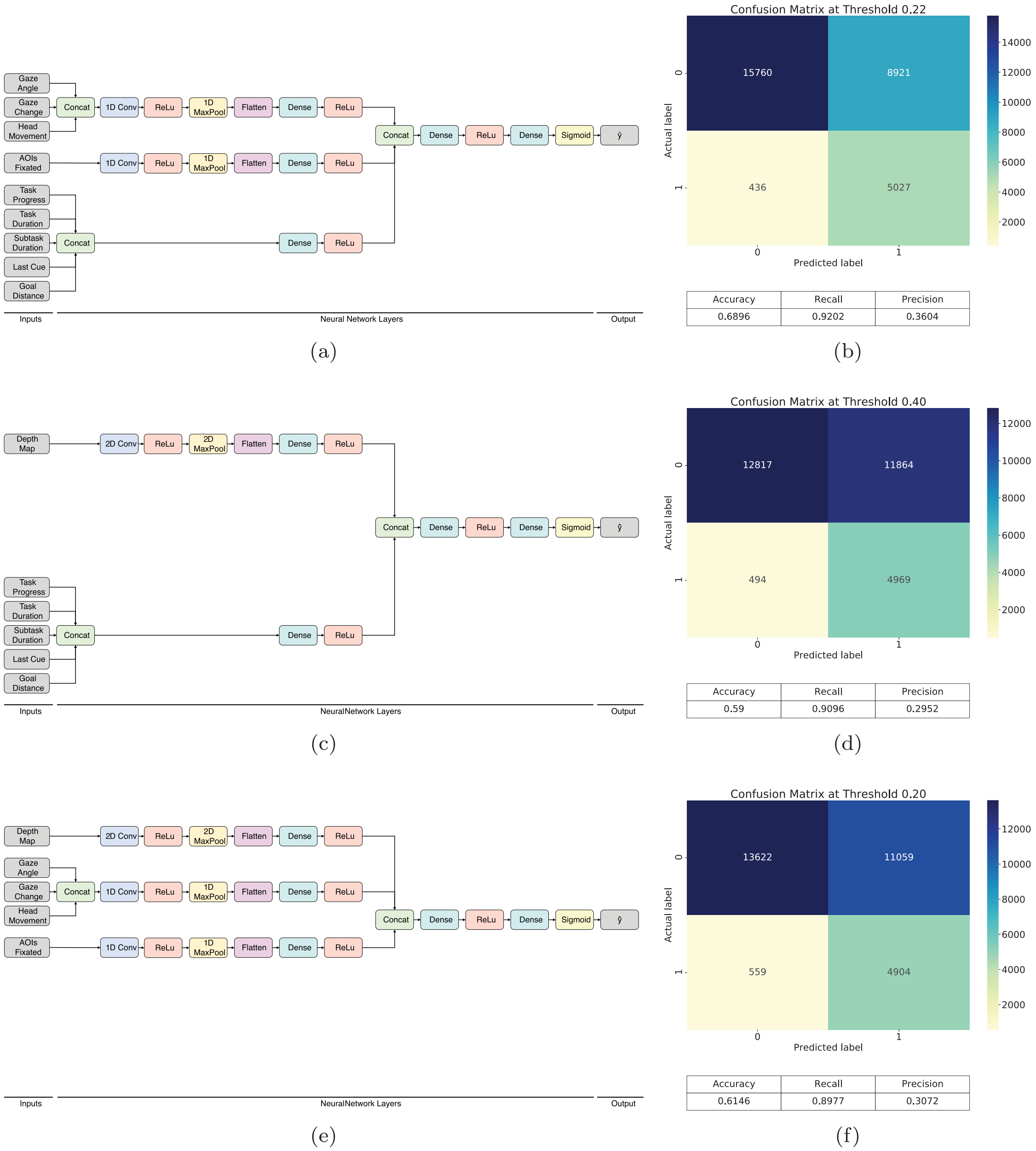
Table B1. Search grid for hyper-parameter tuning and chosen values.

Hyper-parameter	Search grid	Chosen value
Sliding window size (*k*)	[50, 100, 150]	100
Learning rate	[0.0005, 0.0001, 0.00001]	0.0005
Number filters 2D convolution	[20, 40]	40
Kernel size 2D convolution	[(2,2), (4,4)]	(4, 4)
Pooling window size 2D MaxPooling	[(2,2), (4,4)]	(4, 4)
Number of convolution and MaxPooling layers	[1,2]	2
Number hidden units for depth branch	[6, 20, 34]	20
Number filters 1D convolution	[20, 40]	40
Kernel size 1D convolution	[3, 18, 33]	3
Pooling window size 1D MaxPooling	[2, 4]	2
Number hidden units for gaze/head branch	[3, 13, 23]	13
Number hidden units for AOI branch	[3, 13, 23]	23
Number hidden units for task/time branch	[5, 20]	5
Number hidden units for combined branch	[15, 30]	15

Appendix C. Results of sensitivity tests

We conducted sensitivity tests by training the neural network with fewer predictors. Specifically, three configurations were tested, each leaving out one of the three main logical branches of the model: (1) the depth map branch, (2) the head-and-eye movement and AOI branch, and (3) the task and time branch.

The same hyper-parameter optimization as discussed in Section Section 3.3.3 was used, except for the sliding window, which was set to the same size (100) as the final model. Figure C1 shows each of these architectures next to the predictive performance on the test data.



(a)

(b)

(c)

(d)

(e)

(f)

Figure C1. Model architectures and performance without depth map branch (a,b), without head-and gaze movement and AOI branch (c,d), and without task and time branch (e,f).

Appendix D. Experimental room layout

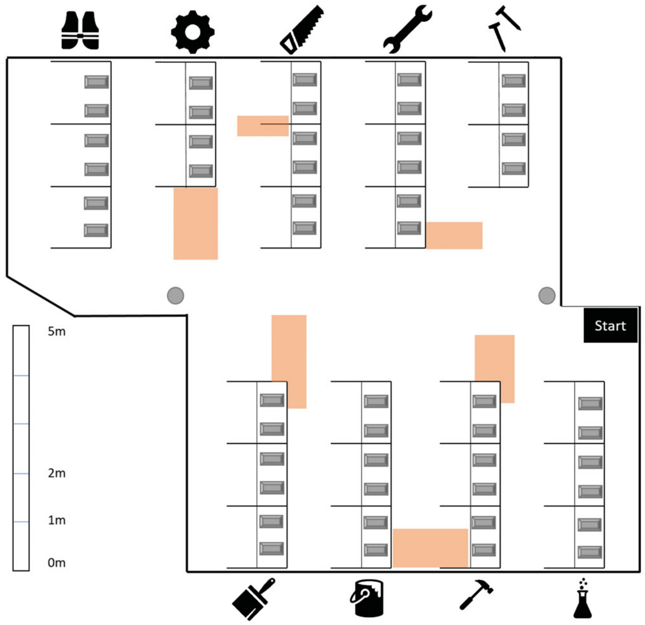


Figure D1. Room layout with zones marked as hazardous in orange and the symbol assignment for each isle of cubicles for User Study 1 block 1.

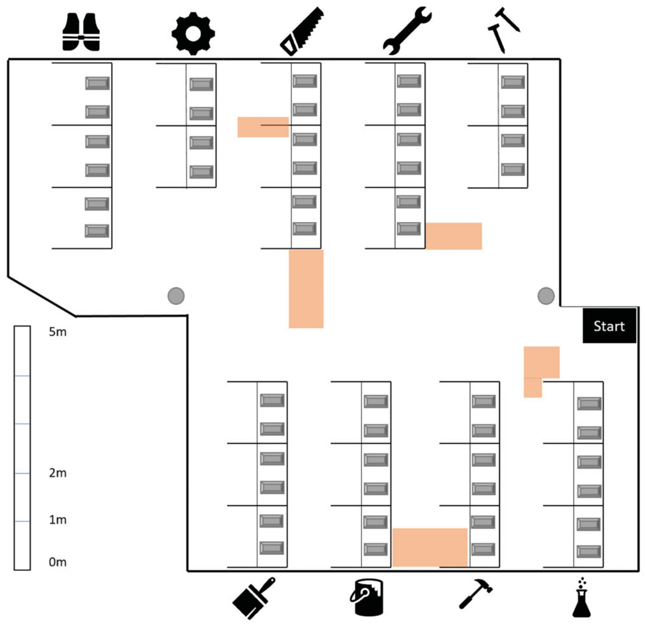


Figure D2. Room layout with zones marked as hazardous in orange and the symbol assignment for each isle of cubicles for User Study 1 block 2.

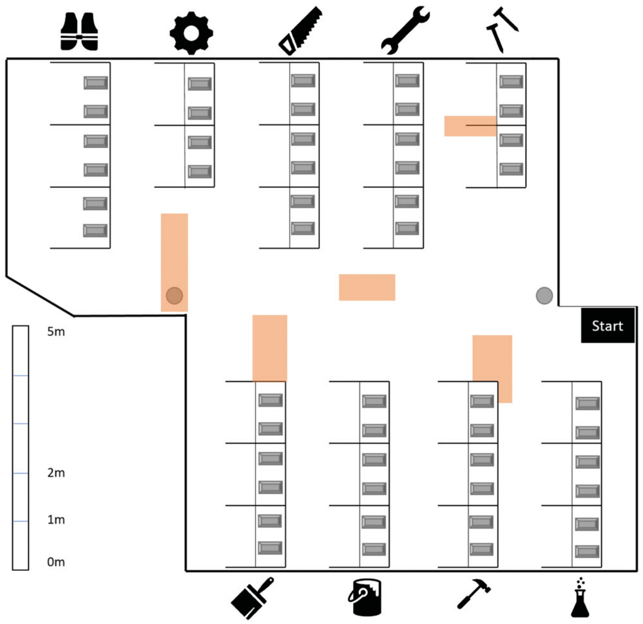


Figure D3. Room layout with zones marked as hazardous in orange and the symbol assignment for each isle of cubicles for User Study 2 block 1.

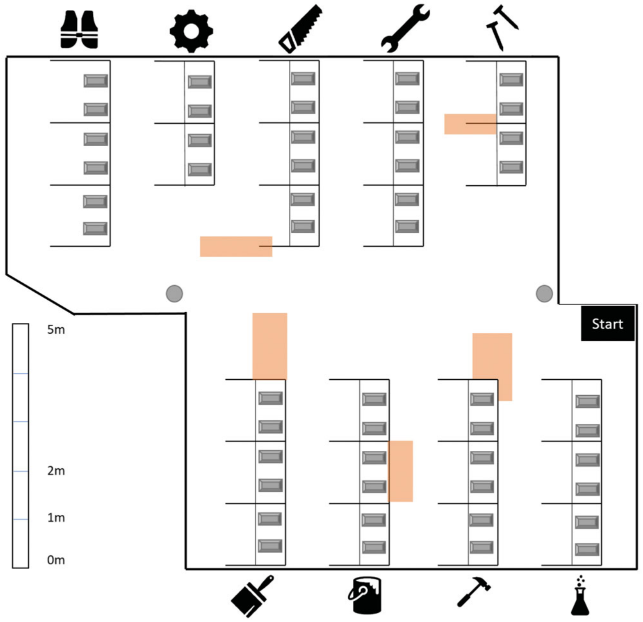


Figure D4. Room layout with zones marked as hazardous in orange and the symbol assignment for each isle of cubicles for User Study 2 block 2.

Appendix E. Feature importance of machine learning model

To investigate model interpretability, we calculated SHAP values (Lundberg & Lee, 2017) for our trained neural network. We used 800 randomly sampled data points from the validation set (out of which 142 had the label $y = 1$) as background data. We further sampled 1000 data points from the validation data to calculate SHAP values over. Here, we extracted samples that had a different label than the 10 samples before it (prediction switch points). Figure E1 displays the absolute values of the calculated SHAP values summed up for each feature we used as input for the neural network. Fixated AOIs and the distance to the target box had a strong impact on the model prediction. The time since the visual cue was displayed last and features relating to eye gaze had a low impact.

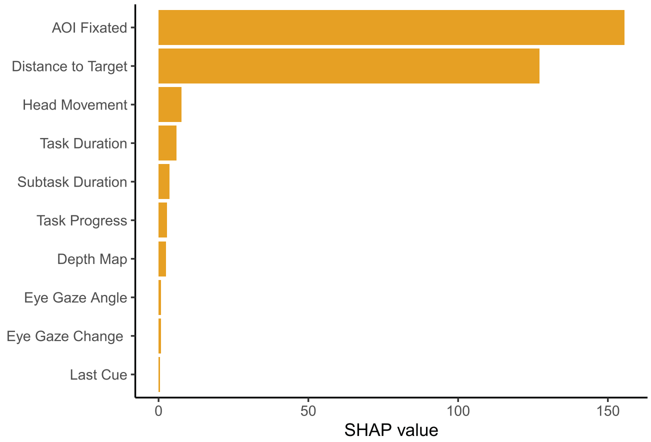
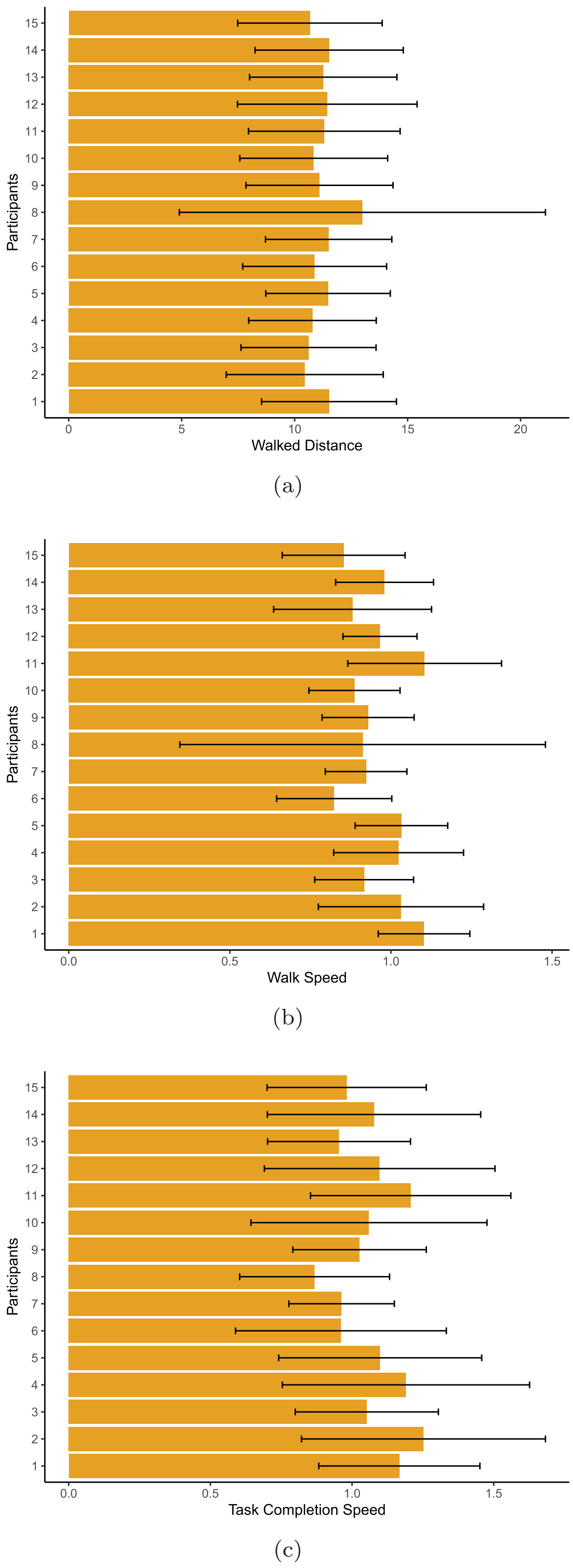


Figure E1. Absolute SHAP values for features of the trained neural network.

Appendix F. Descriptive results of User Study 1

Figure F1 shows descriptive results of each participant in User Study 1.



(a)

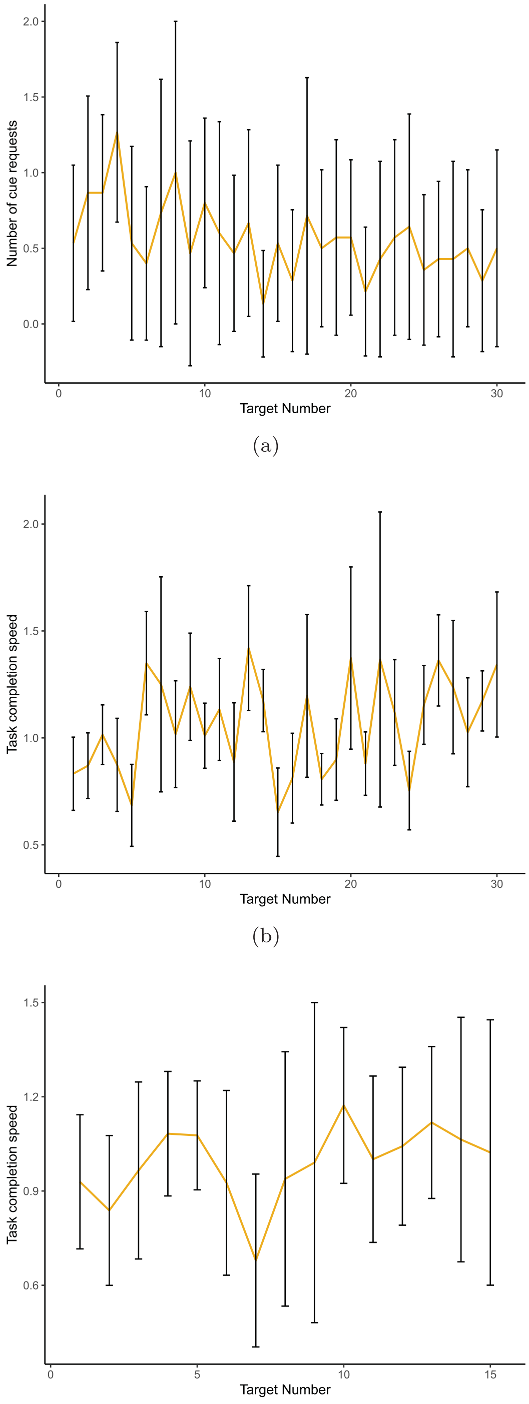
(b)

(c)

Figure F1. Descriptive results of User Study 1: (a) walking distance in meters, (b) walking speed in meters per second, and (c) task completion speed in meters per second for each participant per target. Whiskers show standard deviations.

Appendix G. Exploration of results per target

Figure G1 shows the differences between the targets (i.e., over time) for both User Study 1 and User Study 2.



(a)

(b)

(c)

Figure G1. Exploration of changes over time for both User Study 1 and User Study 2: (a) mean number of activations of the visual cue, (b) mean task completion speed in meters per second per target in User Study 1, and (c) mean task completion speed in meters per second per target of the context-adaptive system in User Study 2. For (a,b) target 1–15 represent the first block of each participant, whereas 16–30 represent the second block (i.e., target 1 is the first target every participant saw in User Study 1). Whiskers show standard deviations.

Appendix H. Statistical analysis of the NASA TLX

Table H1. Wilcoxon signed-rank test results for all NASA TLX dimensions between always-on system and adaptive system in User Study 2.

NASA TLX	V-statistic	*p*-Value
Mental demand	34.0	0.19
Physical demand	15.0	0.93
Temporal demand	21.5	0.22
Overall performance	11.5	0.39
Effort	31.0	0.08
Frustration	19.5	0.39

Appendix I. Correlation analysis of the NASA TLX

Table I1. Pearson correlation coefficients and *p*-values for NASA TLX dimensions with the number of visual cues shown per participant in User Study 2.

NASA TLX	Corr. Coef.	*p*-Value
Mental demand	−0.34	0.34
Physical demand	−0.6	0.07
Temporal demand	0.032	0.93
Overall performance	−0.22	0.55
Effort	0.08	0.83
Frustration	−0.033	0.93