AdapTUI: Adaptation of Geometric-Feature-Based Tangible User Interfaces in Augmented Reality

FENGMING HE*, Purdue University, USA
XIYUN HU*, Purdue University, USA
XUN QIAN, Google Research, USA
ZHENGZHE ZHU, Purdue University, USA
KARTHIK RAMANI, Purdue University, USA

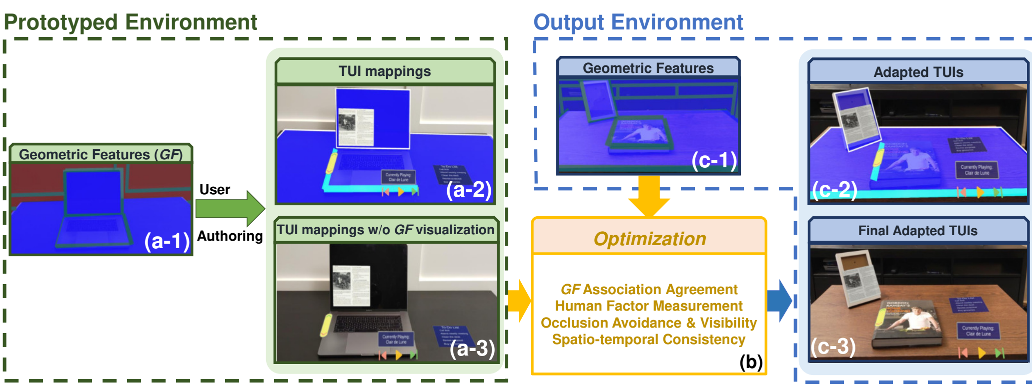

Fig. 1. AdapTUI provides an adaptive solution for geometric-feature-based TUI mappings. (a-1) Leveraging geometric features including edges and planes in the physical environment, (a-2, a-3) a user first prototypes TUI mappings based on their preferences. Later, when the user enters a new environment, (b) AdapTUI utilizes an optimization model to automatically adapt TUI to (c-1) the geometric features in the current scene. (c-2) AdapTUI aligns TUIs with similar geometric features while also maintaining users' ergonomic affordances. (c-3) With AdapTUI, users are able to seamlessly enjoy the TUI layouts in the new environment.

With the advents in geometry perception and Augmented Reality (AR), end-users can customize Tangible User Interfaces (TUIs) that control digital assets using intuitive and comfortable interactions with physical geometries (e.g., edges and surfaces). However, it remains challenging to adapt such TUIs in varied physical environments while maintaining the same spatial and ergonomic affordance. We propose AdapTUI, an end-to-end system that enables an end-user to author geometric-based TUIs and automatically adapts the TUIs when the user moves to a new environment. Leveraging a geometry detection module and the spatial awareness of AR, AdapTUI first lets users create custom mappings between geometric features and digital functions. Then,

*Both authors contributed equally to this research.

Authors' Contact Information: Fengming He, Purdue University, West Lafayette, USA, he418@purdue.edu; Xiyun Hu, Purdue University, West Lafayette, USA, hu690@purdue.edu; Xun Qian, Google Research, Mountain View, USA, me@xun-qian.com; Zhengzhe Zhu, Purdue University, West Lafayette, USA, zhu714@purdue.edu; Karthik Ramani, Purdue University, West Lafayette, USA, ramani@purdue.edu.

ACM 2573-0142/2024/12-ART527
https://doi.org/10.1145/3698127

AdapTUI uses an optimization-based adaptation framework, which considers both the geometric variations and human-factor nuances, to dynamically adjust the attachment of the user-authored TUIs.

We demonstrate three application scenarios where end-users can utilize TUIs at different locations, including portable car play, efficient AR workstation, and entertainment. We evaluated the effectiveness of the adaptation method as well as the overall usability through a comparison user study (N=12). The satisfactory adaptation of the user-authored TUIs and the positive qualitative feedback demonstrate the effectiveness of our system.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; **User interface management systems**; **Systems and tools for interaction design**; **Accessibility systems and tools**.

Additional Key Words and Phrases: Tangible user interface, Augmented Reality, adaptive user interfaces, optimization

ACM Reference Format:
Fengming He, Xiyun Hu, Xun Qian, Zhengzhe Zhu, and Karthik Ramani. 2024. AdapTUI: Adaptation of Geometric-Feature-Based Tangible User Interfaces in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* 8, ISS, Article 527 (December 2024), 26 pages. https://doi.org/10.1145/3698127

1 Introduction

Tangible user interfaces (TUIs) [46] have been utilized to control digital elements while providing users with haptic feedback in Augmented Reality (AR). With the advent of hardware and algorithms, researchers have explored assisting end-users to customize the mappings between AR elements and different tangible inputs, including buttons, knobs, sliders ([8, 17, 35, 37]), and touchscreens [84, 93]. Among the TUI applications, environmental geometric features (***GF***) (i.e., surfaces and edges) have emerged as an essential medium for controlling AR functions because of their ubiquity and easy accessibility [34, 48, 65, 72, 84, 85, 87]. With an adequate amount of *GFs* in the scene, users can enjoy a high degree of flexibility when positioning TUIs on desired *GFs* to ensure their comfort and accessibility. Additionally, the effortless accessibility and comfort facilitated by *GF-based* TUIs confer a substantial ergonomic advantage. By eliminating users' need for actively seeking out a physical object for interaction, *GF*-based TUIs significantly alleviate users' mental burden and cognitive load [14, 86].

Although *GF-based* TUIs enable users to prototype TUIs with fewer restrictions, few systems enable automatic TUI adaptations for users, and users still need to manually reconfigure the mapping between *GFs* and corresponding AR functions every time their surrounding environment changes, which are time-consuming and mentally burdened for users[25, 40, 84]. Meanwhile, the variability of *GFs* makes it challenging to create a universally applicable mapping in different environments. For example, Alex sits in front of his workstation in the office and assigns the following mapping based on his personal use preference (Figure 1a): use 1) the computer's left edge to turn pages for the AR document, 2) the computer's bottom edge to manipulate an AR music player, 3) the desk surface to check an AR to-do list. When Alex works from home and wants to utilize the *GFs* detected in his living room to control the same AR contents, his custom TUIs created in office cannot be directly transferred to the new environment without manually re-binding TUIs and GFs. The number of available *GF* and their positions in the living room are quite different from the office and the original mapping does not exist in the new scene. Given many *GFs* to choose from in the new scene, it would be time-consuming and cumbersome for Alex to manually choose the optimal *GF* for all the TUIs. Further, as the *GF* layout in the new scene is completely different from the original scene, it is challenging to automatically select the appropriate *GF* so that Alex can seamlessly interact with TUIs with comfort and ease of access. Hence, how to empower *GF*-based TUIs to automatically adjust the spatial attachment in different deployment environments while maintaining the same spatial and ergonomic affordance is a crucial open problem in the TUI research.

Some prior TUI applications leverage distinct fiducial markers to register the mapping between digital functions and physical objects, allowing users to transfer mappings by attaching the same marker to various objects [8, 13, 16]. However, this approach has a fatal limitation that it requires additional setups in the deploying environment to complete the adaptation. Alternatively, vision-based techniques have been used to search for physical objects with similar shapes or affordances as the original content [39, 93]. While the vision-based approach mitigates the necessity for additional setups, it requires a high degree of similarity in both shape and size between the original object and the object that appeared in deploying environment, which may not always be feasible.

In light of the limitations of the marker-based methods and vision-based methods in TUI adaptation, researchers have explored optimization-based approaches that provide greater flexibility and do not require the same object or marker to appear in the deploying scene. Leveraging spatial information from different scenes, optimization-based methods successfully generate layouts for general UIs in various physical environments [9, 20, 51, 57, 89]. Unfortunately, the optimization methods have failed when they are applied to *GF*-based TUI applications. Targeting general UIs without tangible interaction, these works take the spatial relationship between AR functions and physical assets as the primary optimization basis, and maintain consistent spatial affordance by looking for an object of the same category as the original UI-attached object. Although the optimization methods looking for physical objects of the same category do not necessitate physical objects to share high similarity as vision-based methods, they remain incapable of dealing with the issue when physical objects of the same category are absent/duplicated in the new scene. Therefore, current optimization methods cannot be directly applied to *GF*-based TUI adaptation. Moreover, the optimization methods focusing on general mid-air UI adaptations overlook crucial ergonomic factors such as users' comfort and accessibility, which are indispensable for effective TUI adaptations. To illustrate, in the scenario of Alex's living room, the AR document originally mapped to the monitor may be re-mapped to TV, if the system simply looks for the object with the same category. Nonetheless, such mapping may not be convenient for Alex; Alex has to walk to the TV from his current position to interact with the TUI. Additionally, looking for the same category of objects heavily relies on object recognition [9, 51, 57, 89], which easily fails when the deploying environment does not contain the original category of objects. More importantly, vanilla object recognition operates on object level and therefore cannot be directly used for choosing appropriate GFs.

While we fully recognize the advantages of the optimization-based approaches in terms of adaptation systems, current optimization methods cannot be applied to *GF*-based applications and do not consider users' comfort. To this end, we are motivated to take users' ergonomics affordance into consideration and develop a framework that adapts *GF*-based TUI mappings using optimization computing.

We propose AdapTUI, an optimization-based solution that adapts the geometric-based TUI to different environments while maintaining both ergonomic and spatial affordances between GFs and AR functions. With AdapTUI, a user can initialize their preferences for TUI placement in the original scene, and when the user enters the new scene, our system automatically adapts the TUI layout so that users can comfortably and easily view, modify and interact with the TUIs. We explore the design goals for *GF*-based TUI adaptations and summarize four sub-objectives for adaptation process, including consistent geometric features, maintaining user-centric ergonmomic affordances, visibility and spatio-temporal consistency. Driven by the adaptation objectives, we develop an optimization-based approach for the automatic adaptation process that considers a set of factors as input: 1) TUI characteristics, 2) GFs in the environment, and 3) users' information. In summary, we highlight our contributions as follows:

- An optimization-based algorithm that leverages spatial information and ergonomic factors so that end-users can comfortably and easily interact with *GF*-based TUIs in different environments.
- An integrated system for allowing end-users to author TUIs with original mappings between GFs and AR functions, and automatically adapting the attachment of user-authored *GF*-based TUIs in the new environment while ensuring the same spatial and ergonomic affordances.
- A set of UI applications that highlights the versatility of AdapTUI in diverse physical environments.

2 Related Work

2.1 Adaptive TUIs in AR

Tangible user interfaces (TUIs) leverage physical objects to provide haptic feedback and spatial reference so that end-users can opportunistically and accurately manipulate AR functions [40, 54]. To enable the usage of TUI in different scenarios, researchers have investigated various adaptive TUI strategies, including ***marker-based***, ***tracking-based***, and ***gesture-based*** methods.

Marker-based TUIs use instant tangible carries to build the connection between the physical environment and digital assets. With fiducial markers, end users can utilize different physical objects to control the AR functions by switching the marker-attached objects [8, 13, 16, 17, 30, 40, 49, 54, 79, 82, 94]. Further, users can instantly access and interact with the connected digital content when the reference markers are present in the target scene. Yet, markers require additional preparation and may also additionally occupy the interaction area, which slackens users' manipulation of physical objects.

Tracking-based TUI utilizes RGBD cameras or hardware sensors to track the real-time status of physical objects and maps TUIs with tracked objects. Typically, vision-based tracking systems utilize features of physical objects to identify the pose [12, 71] or 6 Degree-of-Freedom (DoF) [17, 25, 36, 39] of the target object. On the other hand, the hardware sensor-based systems [56, 69, 75, 91] place sensors on the object surface or inside area to acquire the object 6DoF information and detect users' interaction with TUIs. Although *tracking-based* TUIs resolve the problem of complicated settings and extra occupation compared with the *marker-based* approaches, they require that the same item always appears in the current scene. Therefore, users have to manually assign the mapping to a new object if the original physical is missing in the new scene.

The latest *gesture-based* adaptation systems loosen the constraints of the same content in the new scene. *gesture-based* adaptation systems claim that users naturally perform different gestures when interacting with objects of different shapes, like holding a cup versus grasping an apple [3, 93]. Therefore, these systems decide the TUI mapping target object by detecting users' performed gestures. As users can use similar gestures (e.g. holding [93]) with different objects, it allows users to interact with a different object even if the original object is missing in the new scene. However, they still require objects in the new scene to have similar shapes to original objects. Moreover, these systems are not able to decide the most proper physical object for rendering TUIs when multiple similar physical objects are present in the new scene. Rather than requiring similar physical objects, OmniTouch [34] and Tailored Controls [2] focus on the low-level geometric feature, planes, and render the corresponding AR content by detecting performed gestures on the planes. Nevertheless, these methods require extra inputs from users, which makes it not fully automatic and would be cumbersome for users if they want to perform multiple tangible interactions (or on different planes).

In most TUI systems discussed above, the adaptation is limited to the hand-held objects [8, 12, 17, 57, 69, 93], which require additional setup (e.g., markers [8, 17]) or the same/similar object to appear in the new scene. Hence, these systems cannot deal with problems of spatial variation and duplication/absence of physical objects in the new scene. Furthermore, even if the same object is available in the new environment, TUI mappings may fail due to potential tracking challenges, including occlusion, background variations, changes in lighting conditions, and different gestures performed. Compared with directly mapping TUIs to a single target object, *GF*-based TUI adaptations face a different challenge, in which the system needs to choose the most appropriate *GF* from a considerable number of *GF*s in the scene. Further, in order to appropriately adjust the TUI attachment while keeping the spatial and ergonomic affordances, the *GF*-based adaptation needs to take environmental and human factors into consideration accordingly.

Due to the stringent requirement for objects' visual similarity, tracking challenges, and lack of consideration of environmental and human factors, current TUI systems are not suitable for adapting *GF*-based TUIs. To this end, we strive to develop an adaptive TUI system that harnesses ubiquitous and easily accessible *GFs* in the physical environment and considers both spatial and human factors as the system adapts. Our system provides an end-to-end workflow that allows users to prototype initial TUI mappings and provides automatic TUI mapping transferring to new *GFs* when the surrounding environment changes, thereby providing a more flexible and adaptable solution.

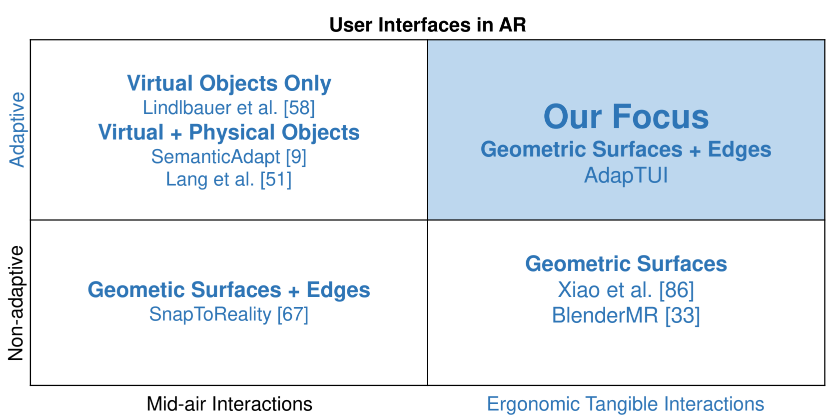

Fig. 2. Related Work of AR User Interface Alignments and Adaptations

2.2 Optimization-Based User Interfaces

Optimization algorithms that consider multiple factors have been proven effective in achieving general UI placements and adaptations [9, 21, 26, 32, 33, 38, 55, 57, 57, 58, 67, 78, 86, 89]. Generally, compared with manual adjustments which are time-consuming and cognitive-burdened, the automated nature of optimization algorithms significantly eases users' burden, especially when there are multiple UI elements to deal with and surroundings change frequently.

Combining rule-based decision-making with optimization computing, Lindlbauer et al.[58] decides the level-of-detail (LOD) of mixed reality (MR) interfaces based on the user's cognitive load. Similarly, Tailored Reality [15] reorganizes physical layouts in MR using an optimization method that emphasizes users' perceptions of different entities. Yet, both of these works focus on users' perceptions and overlook the physical environment, which plays a key role in adapting UIs into different scenes. Researchers have also explored the connection between UIs and the physical environment. SnapToReality [67] extracts the *GFs* in the environment and enables users to easily align AR contents to their desired *GFs*. However, it only supports user-defined UI placement in the current scene and is not able to adapt the user-defined UIs to different physical environments. Niyazov et al. [66] and AUIT [20] enable users to refine optimization constraints for AR content layouts but also introduce the problem that a lot of manual actions are required. SemanticAdapt [9] and Lang et al. [51] adapt Mixed Reality (MR) interfaces to different environments leveraging semantic connections between virtual contents and physical objects. However, these works have a problem in that most of the adaptations are based on object level, which means the systems require target objects to appear all the time and adaptations could easily fail when target objects are missing in the new environments. More importantly, compared to TUI adaptations, where the tangible input is bound to the physical entities and users need to interact with the physical entities in-situ, these UI adaptive works primarily consider the mid-air interactions and neglect users' ergonomic affordances during interaction with physical environments. Rather than focusing on mid-air manipulations, recent works [10, 33, 86] enable users to perform tangible interactions leveraging physical affordances. Xiao et al. [86] and BlendMR [33] enable users to perform tangible interactions on physical surfaces. Different from our work concentrating on adaptations, these works focus on how users interact with the UI and how UIs seamlessly blend with the physical objects, respectively. InteractionAdapt [10] adapts UIs in VR-based workstation scenarios leveraging users' nearby physical affordances. Yet, the work is targeted at the workspace in VR and ignores crucial visual feedback from the physical objects/environments.

While acknowledging the advantages of computing optimization, we notice that most current optimization works like, SemanticAdapt [9] and InteractionAdaption[10], either focus on mid-air UI adaptations and therefore overlook users' ergonomics, or ignore the complexity of the physical environments, which are important factors in TUI adaptation computing. Further, most of the current optimization works [9, 20, 51, 66] maintain the spatial affordance of UIs by seeking an object in the same category as the original UI-attached object and rearranging UI elements with the physical object. This process requires the presence of objects of the same category in both the original and new environments, rendering the system incapable of addressing scenarios involving objects' duplication/absence, similar to previously discussed object-based TUIs in Sec 2.1. Compared to mapping UIs exclusively to the sole existing object, *GF-based* TUI adaptations aim to choose the most appropriate *GF* given a substantial amount of *GFs* in the scene. To this end, the optimization schemes differ between object-based adaption and *GF*-based adaptation, rendering current computing optimization methods unsuitable for direct application to *GF-based* TUI adaptations. Compared to prior mid-air UI adaptations that either neglect users' ergonomics or rely on the sole presence of objects, AdapTUI targets *GF*-based TUI adaptations, which eliminates the necessity for physical objects and ensures the preservation of spatial and ergonomic affordances during adaptations. Our system uses an optimization-based algorithm that incorporates both physical environments and ergonomics as inputs. In addition to general surface-based UIs (like Xiao et al. [86], and InteractionAdapt [10]), we also explore adaptions based on geometric edges, which provide sharp tactile feedback for users [35, 47].

2.3 Ergonomics in Tangible Interactions

Ergonomics play an essential role in the design of 3D virtual interfaces, especially when tangible interactions are taken into consideration [18, 63, 64]. Make it Home [90] designed a virtual furniture placement system that makes sure that the virtual furniture is placed in a proper position and orientation for easy access. Similarly, AUIT [20] considers reachability as a primary consideration for mid-air UI layout. Besides accessibility, which is related to virtual content's global position, arm fatigue, which is related to virtual content's local placement, is another important factor [41, 52]. Erg-O [63] claims that a comfortable location is crucial for virtual content placement and the virtual content placement should be tailored to different users. For example, the virtual content will not be placed on the right-hand side for a left-handed user. Recently, XRgonomics [19] proposes that ergonomics should be taken into consideration for MR interface design. To facilitate this, XRgonomics designs an MR toolkit that enables visualization of ergonomic metrics and interaction costs. Yet, most works do not consider tangible interactions with physical objects. In terms of tangible interactions, Magic Desk [4] explored the comfortable zone of tabletop surfaces for multi-touch interactions. Cheng et al. [11] investigated the impact of physical tables for interactions in virtual reality (VR) in terms of ergonomics.

Inspired by these works, we incorporate ergonomics factors, including accessibility and arm fatigue, into the design of optimization schemes. Despite considering the ergonomic factors, we also apply common factors, like visibility and spatio-temporal consistency [9, 20, 61] into our algorithm. Therefore, while maintaining spatial affordances among different scenes, AdapTUI is also able to effectively keep users ergonomic when interacting with TUI applications.

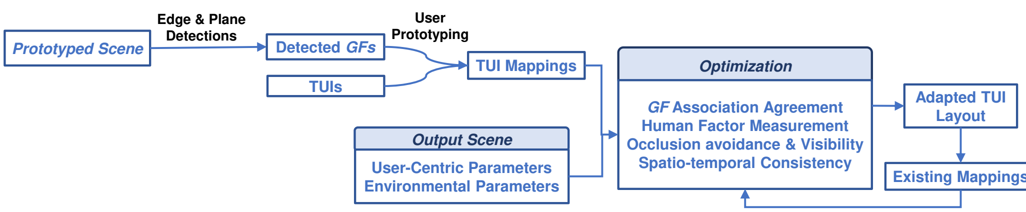

Fig. 3. Overall workflow of AdapTUI.

3 TUI Mapping Optimization

Given a set of TUI elements, the *GFs* on the physical objects, the user's status, and prototyped TUI mappings in the ***prototyped scene***, for each TUI, our goal is to determine its placement location automatically in the current MR environment (***output scene***) primarily based on spatial and ergonomic affordances. Specifically, the placement location involves three parameters: 1) visibility: whether the TUI element should be visualized in the current scene, 2) connected GF: the most suitable *GF* to connect with the TUI element, and 3) local position: the specific location of the GF where TUI element will be mapped to. To achieve the placement goal, we design a combinatorial optimization method that first determines the visibility, then chooses connected *GF*, and finally decides the local position of *GF*. In this section, we first describe the adaptation objectives. Next, we introduce the inputs/parameters for our optimization method. Finally, we will discuss the optimization cost functions and constraints for computing.

3.1 Adaptation Objectives

We derived the following adaptation objectives (**AO**) where the motivation has been extensively discussed in the Related Work Section.

- **AO1: Consistency in geometric spatial affordances.** To keep consistency in the spatial relationship between TUI and *GFs*, we need to ensure that TUIs are mapped to *GFs* with similar inherent properties. For example, a TUI attached to a vertical plane (e.g. monitors) should not be mapped to an edge or a horizontal plane (e.g. tables) in the new environment. In addition to the inherent geometric properties, the spatial relationship between *GFs* and virtual contents should be stable as well.

- **AO2: Maintaining user-centric ergonomic affordances.** TUI applications should be positioned in an accessible and comfortable zone that tailors to users' preferences, as discussed in Sec 2.3. For instance, if users prefer using their left hand to interact with an AR speaker's TUI while seated, they may not appreciate a layout where the speaker is far away or positioned on their right side.
- **AO3: Layout visibility and occlusion avoidance.** Our system renders TUIs in the *output scene* according to whether appropriate *GFs* are found. If there is no appropriate *GF* that satisfies the spatial and ergonomic requirements, the TUI will remain hidden in the *output* scene, similar to previous work [9, 20, 58]. On the other hand, if a suitable *GF* is available, the TUI will be mapped to an available space on the *GF* and can be resized as needed to prevent occlusion from other TUIs/physical objects.
- **AO4: Spatio-temporal consistency.** In the current adaptation, the TUI adjustments made during the previous round will be given higher priority to avoid redundant updates. In detail, the adaptation process will only be updated when there are substantial changes in the environment.

3.2 Inputs

Driven by the above AOs, our method considers three main sources of input: ***TUI Element parameters*** that hold coherent characteristics of the AR application, ***Environmental Parameters*** that contain information in the *prototyped scene* and *output scene*, and ***User-centric Parameters*** that refers to users' status in both *prototyped scene* and *output scene*. We summarize the input parameters in Table 1.

3.2.1 TUI Element Parameters. Our system computes and records each TUI element's inherent or user-defined properties and takes these properties as optimization model input. Different form UI which integrates the input and output in the same location, some TUI applications may have the tangible input and the AR output in separate locations, e.g. an edge-based tangible input to control an AR audio player output floating in the air. In our system, we consider optimization for tangible input and AR output layout respectively if they are separated (d_e), where the TUI element is specified as input or output through i_e. We define the $i_e = 0$ for tangible input and integrated TUIs, and $i_e = 1$ for separate tangible output. While the AR output may stay in mid-air as other MR user interfaces, the tangible input needs to be attached with a *GF*. Our system records the attached *GF* and the mutual connection between the element and its corresponding input/output in the a_e. In other words, a_e contains all elements that are related/associated with the element e. Similar to prior work [9, 20], the position p_e is used to optimize for spatio-temporal consistency and scale s_e is used to constrain the element's position and avoid overlaps in the *output scene*. The TUI type represents the dimension of the AR element, where 0 refers to a button/slider-like AR element and 1 refers to 2D/3D interfaces.

3.2.2 Environmental Parameters. The information obtained from the *prototyped scene* and the *output scene* contributes to the final layout of TUI elements. For each scene, we consider all *GFs* denoted as G. Note that an object may contain multiple GFs, e.g., an empty table has a plane and four edges. For each GF, $g \in G$, we use the GF type t_g to classify whether the GF is an edge or plane, where 0 refers to the edge type and 1 refers to the plane type. The position p_g and size s_g of GFs will be used for localizing the specific position of the TUI element with respect to the geometric feature. For example, users may like to tie the TUI input at the center of an edge. If the current edge does not have enough length for the TUI, the TUI may be scaled to the new edge's length. Further, we used the rotation r_g to present the status of GF that users may interact with. For example, the

Table 1. Description and ranges of input parameters

TUI Element Parameters	
Parameter	Description
$E = (e_1, ..., e_n)$	All TUI elements
$N_e \in \mathbb{Z}^+$	Number of TUI elements
$d_e \in \{0, 1\}$	Whether tangible input and AR function are separated for element e
$i_e \in \{0, 1\}$	Whether the TUI element is the input or output
$t_e \in \{0, 1\}$	Type of the TUI element e
$p_e \in \mathbb{R}^3$	Position of element e in the *prototyped scene*
$s_e \in \mathbb{R}^3$	Scale of element e in the *prototyped scene*
$r_e \in \mathbb{R}^3$	Rotation of element e in the *prototyped scene*
a_e	associations of element e
Environmental Parameters	
Parameter	Description
$G = (g_1, ..., g_n)$	All geometric features in the scene
$N_g \in \mathbb{Z}^+$	Number of GFs in the scene
$t_g \in \{0, 1\}$	Type of the geometric feature g
$p_g \in \mathbb{R}^3$	Position of geometric feature g
$s_g \in \mathbb{R}^3$	Size of geometric feature g
$r_g \in \mathbb{R}^3$	Rotation of geometric feature g
User-centric Parameters	
Parameter	Description
$J = (j_1, ..., j_n)$	User's body elements
$N_j \in \mathbb{Z}^+$	Number of joints
$p_j \in \mathbb{R}^3$	Position of user's body joint j
r_h	Rotation of user's head

user would like to align an AR photo frame on a wall plane rather than a horizontal plane. Here the r_g refers to the line vector for edges and refers to the normal vector for planes.

3.2.3 User-Centric Parameters. In addition to the TUI characteristics and information from the environment, we also take the users' status into account, including the user's upper-body joint, position, and rotation of the body. As TUI only requires hand interaction with the user, to reduce the computation complexity, we only consider the upper-body joints (J) which are related to the touch interaction. In addition to the hand joint, the arm and shoulder joints are taken as input because these joints significantly affect the ergonomics like fatigue [19, 52]. We used inverse kinematics (IK) [80] to automatically get positions for the arm joints p_g. What's more, we also consider the user's head movement including position and rotation (r_g) in the adaptation to make sure the TUI is within the user's field of view and reachable area.

3.3 Adaptation Schemes

Our optimization method lies the decision on whether TUI element e will be aligned with GF g, which is represented as

$$x_{e,g} = \begin{cases} 1 & \text{if e is aligned with g} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

As mentioned in Section 3.1, we mainly consider four adaptation objectives (*AOs*). Based on the *AOs*, we derive the object consisting of four sub-objectives that respectively refers to the *AOs*: (1) GF association agreement (A) that measures whether the TUI is mapped to a GF that holds similar characteristics as the original one (*AO1*), (2) human factor measurement (H) evaluates whether users can access the TUI easily and comfortably (*AO2*), (3) occlusion avoidance and visibility (O) represents the measurement of whether there is a collision with other TUI items and whether the object is occluded by other physical objects in the scene (*AO3*), and (4) spatio-temporal agreement (S) which is a measure of consistency between the previous and current adaptation results. We formulate our object function as a weighted sum of the sub-objective terms:

$$\max_{e,g} w_a \cdot \hat{A} + w_h \cdot \hat{H} + w_o \cdot \hat{O} + w_s \cdot \hat{S} \tag{2}$$

where $w_a + w_h + w_o + w_s = 1$ and we empirically set the weights to be $w_a = 0.3$, $w_h = 0.3$, $w_t = 0.2$, and $w_s = 0.2$. After determining the mapped GF g for a given e, our system localizes the TUI with respect to the GF referring to the relative position, rotations, and scales in the *prototyped scene*.

3.3.1 GF Association Agreement (A). The GF association agreement relies on the suitability of assigning a TUI element e to a geometric feature g in terms of geometric feature status and TUI characteristics (*AO1*). Based on this sub-objective, we place the TUI on *GFs* with similar inherent characteristics and spatial connection, which are represented by association (a_e) and *GF* parameters. We first use the Euclidean distance $\Delta_{e,g}$ to present whether the type of TUI-mapped *GF* (t_e) is the same as geometric feature's type t_g. We reward when the association and *GFs* share similar type attributes. For the *GF* factors, as the position attributes are more related to human factors that we will discuss later, the rotation factors reflect more about the *GFs* self. We compute the angle difference between the current *GF* and original *GF* on the vertical (y) axis, which is denoted as θ_y. Additionally, we compute the angle difference between *GFs* and original **GF** in forward(z) and right direction(x), which are denoted as θ_z and θ_x respectively. Note that these two angle differences are computed in the user's local coordinate. Overall, we compute the GF Association agreement as

$$A = \frac{1}{N_e} \sum_e \sum_g (w_{a_1}(1 + e^{\Delta_{e,g}})^{-1} + w_{a_2} e^{-2\theta_y{}^2} + w_{a_3} e^{-\theta_x{}^2} + w_{a_4} e^{-\theta_z{}^2}) \tag{3}$$

where weights are empirically determined $w_{a_1} = 0.5$, $w_{a_2} = 0.3$, $w_{a_2} = 0.3$, $w_{a_3} = 0.15$, $w_{a_4} = 0.15$.

3.3.2 Human Factor Measurement (H). We aim to generate layouts that are accessible and comfortable for users to interact with (*AO2*). Therefore, we mainly focus on accessibility and ergonomics in the human factor measurement. In detail, for accessibility, we reward places that are within users' reach and at the same time in front of the user. We compute the distance between the user's joints and the *GF* horizontally (projection on XZ plane) and vertically (y axis), denoted as $lxz_{u,g}$ and $ly_{u,g}$. We define a function H_{r_1} and H_{r_2} that rewards when the object is within users' reach

$$H_{r_1} = max(0, lxz_{u,g} - w_{l_1} \cdot l_{arm}) \tag{4}$$

$$H_{r_2} = max(0, ly_{u,g} - w_{l_2} \cdot l_{arm}) \tag{5}$$

where $w_{l_1} = 0.72$, $w_{l_2} = 0.18$ and l_{arm} refers to users' arm length. We also reward when objects are in front of users or at the sides of users

$$H_{dir} = \begin{cases} 0 & \text{if the object is behind the user} \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

We combine the H_{r_1}, H_{r_2} and H_{dir} as

$$H_r = w_{r_1} \cdot e^{-H_{r_1}} + w_{r_2} \cdot e^{-H_{r_2}} + w_{dir} H_{dir} \tag{7}$$

where $w_{r_1} = 0.2$, $w_{r_2} = 0.2$ and $w_{r_3} = 0.15$. Meanwhile, we utilized Rapid Upper Limb Assessment (RULA) [62] to measure the upper body ergonomics scores. RULA is a widely accepted ergonomics measurement that assigns posture scores to joint angles and can be used for different postures. A high RULA score refers to an uncomfortable/low-ergonomic status. We compute the RULA score $R(e, g, j)$ for arm joint angles and penalize positions where RULA scores are high. Similar to Erg-O [63], we interpolate joint angles and RULA scores to avoid sudden score changes. In summary, we define the human factor measurement as

$$H = \frac{1}{N_e} \sum_e \sum_g x_{e,g}(H_r - w_r \cdot R(e, g, j)) \tag{8}$$

where $w_r = 0.45$.

3.3.3 Occlusion Avoidance and Visibility (O). The sub-objective of occlusion avoidance penalizes the placement where TUI objects are occluded by other physical or existing TUI layouts *AO3*. To achieve this, we choose a set of sample points (pt) on the geometric feature and use raycast [73] from the user to the sample points to check whether the *GF* is occluded by other objects. We define a binary function that reflects whether the occlusion happens for each sample point pt:

$$x_{g,pt} = \begin{cases} 0 & \text{if the raycast hits the } GF \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

The ultimate sub-objective is defined as

$$O = -\frac{1}{N_e N_{pt}} \sum_e \sum_g \sum_{pt} x_{e,g} \cdot x_{g,pt} \tag{10}$$

where the N_{pt} refers to the number of sample points for each tangible elements. The N_{pt} is emperically set as 16.

3.3.4 Spatio-Temporal Agreement (S). The spatio-temporal agreement promises consistency between prior adaptation results and the current layout, avoiding unnecessary updates until significant changes occur in the environment or users' status *AO4*. In the implementation, we mainly consider the existence of TUI for a GF ($x_{e,g}$) in the current environment and user's status changes. Specifically, we calculated the local position of *GF* $\delta_{u,g}$ with respect to users and then record $\delta_{u,g}$ every time the system updates. Then we define the $\Delta_{u,g}$ that reflects the user's move during the update

$$\Delta_{u,g} = \begin{cases} 0 & \text{if} \quad ||\delta_{u,g} - \delta^*_{u,g}|| < 0.5 * l_{arm} \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

where $\delta^*_{u,g}$ is initialized as the corresponding local position in the *prototyped scene* and then updated with the last recorded position during the optimization. We then formulate the sub-objective as

$$S = -\frac{1}{N_e} \sum_e \sum_g (1 - x_{e,g}) \cdot \Delta_{u,g} \tag{12}$$

where $\delta^*_{u,g}$ was initialized to the scaled distance at *prototyped scene* and updated with last recorded distance.

3.4 Constraints

In order to ensure an effective optimization formulation, we introduce a set of constraints in addition to the prior sub-objectives and terms. To avoid a TUI element being located in multiple places (*AO1*), we enforce

$$\sum_{g} x_{e,g} \leq 1, \forall g \tag{13}$$

To avoid a TUI element being aligned with a *GF* with an undesirable size (*AO3*), we require that the size ratio between *GF* (s_g) and TUI element (s_e) is above an empirical threshold

$$s_g < 0.8 s_e \Rightarrow x_{e,g} = 0 \tag{14}$$

To avoid the incompatibility between TUI dimension and the *GF* type, e.g. a touchscreen is aligned with an edge, we formulate the constraint as

$$d_g < d_e, i_e \neq 1 \Rightarrow x_{e,g} = 0 \tag{15}$$

4 Implementation

In this section, we explain the implementation of AdapTUI which consists of the concrete workflow for using AdapTUI (Figure 3), authoring and adaptation interfaces, and the implementation details.

4.1 Adaptation Workflow and Authoring Interface

With AdapTUI, users are able to first manually prototype their preferences for TUI layouts in the *prototyped scene* and then visualize and interact with automatically adapted TUI applications in the *output scene*. To support users with a convenient prototyping process of TUI applications, we employ a main menu (Figure 4a) that floats next to users' left hand and provides corresponding textual guidance.

During the usage of AdapTUI, users first enter *Data Collection Mode* by clicking the *Start Scanning* button and then walk inside the environment. At the same time, AdapTUI extracts the *GFs* in the scene through edge and plane detection, and visualizes the detection results in the AR scene (Figure 4c). To recognize positions and rotations of indoor planes (including walls, planes, and object surface planes), a real-time plane detection algorithm, PlaneRCNN, [59] is employed. Meanwhile, for 3D line detection, a real-time line segmentation algorithm [28] is used. A lidar camera [45] is used for capturing the RGBD information of planes and edges, and the detection results are rendered in AR world with the support of HoloLens2's SLAM. After confirming the critical GFs are correctly detected, the users can press the *Save Scene* button. This operation tells AdapTUI to stop edge/plane detection for the environment and to save the information of *GFs* in the system. After completing the *GFs* extraction, the users can click the checkbox of *Show GFs* to toggle on/off the visualization for *GFs*. Then the users start the personal mapping for TUIs in the *prototyped scene* through *Author Mode*. Users can choose desired TUI application from a TUI library and align the TUI with their desired GFs by moving the TUI near desired *GFs*. To ensure users are connecting TUIs with desired *GFs*, AdapTUI highlight *GFs* when a TUI is nearby (Figure 4b-2). Then the users can adjust the TUI position and rotation until they are satisfied using bare-hand interaction, which is supported by HoloLens2. Our system further automatically snaps the TUI to the GF for precise alignment. For instance, an edge-based TUI is adjusted so that its direction is the same as the edge. When the users finish all TUI prototyping, they are able to save the prototyping results and AdapTUI records the TUI information.

When the users enter the new environment, they can click the *Adaptation* button to enter *Adaptation Mode* after the system finishes the *GF* extraction for the current environment. Our system then starts the optimization step and updates only when new *GFs* are detected or users' location is beyond a distance of 0.5 l_{arm} from the last recorded location (see Section 3.3), where l_{arm} refers to the arm's length. In the new environment, our system automatically hides the plane and edge visualization for users to better interact with TUIs without visual occlusions. Additionally, the users can edit the unsatisfying layout or add new TUI layouts for the scene by switching back to *Author Mode*. At the same time, our system will turn on the *GFs* visualization again in the *Author Mode*.

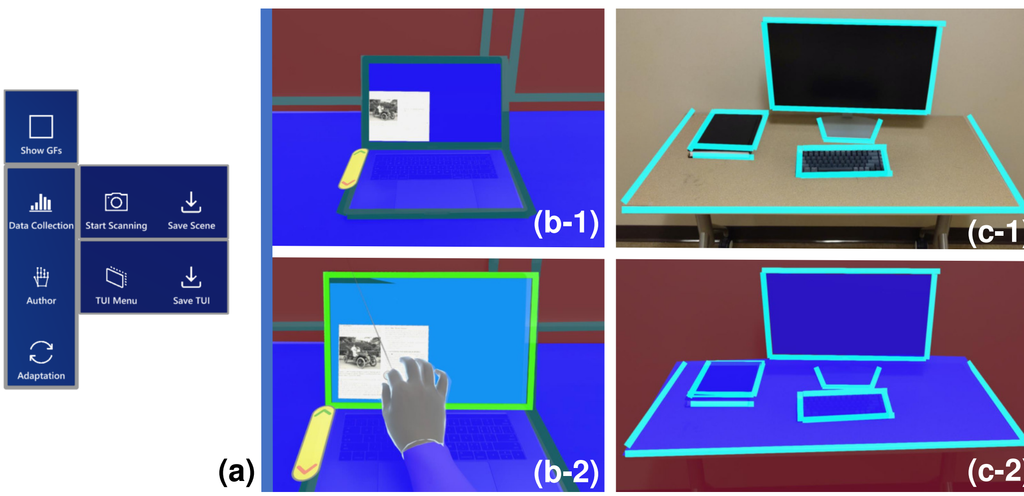

Fig. 4. (a) The left-hand menu for users to collect data in the environment, conduct initial TUI mapping in the *prototyped scene*, and perform optimization adaptation. (b-1, b-2) The *GF* will be highlighted when a TUI is nearby during authoring. (c-1) Sample edge detection and (c-2) plane detection results and their visualizations.

4.2 Software and Hardware Setup

In AdapTUI, to accurately track edges and planes, we adopt an Intel Realsense ™ lidar camera L515 with high depth accuracy ([45]) for geometric feature tracking, and combine it with a HoloLens2 AR-HMD for TUI application usage following prior arts [35]. Specifically, the lidar camera provides plane and edge detection while HoloLens2 provides hand tracking and AR user interfaces. The lidar camera and HoloLens camera were calibrated ahead, and the lidar camera was connected to a backpack computer (HP VR Backpack [43], Intel Core i7-8850H, 2.6GHz, 32GB RAM, NVIDIA 2080 GPU), which provides computation power for both edge detection and optimization. We chose resolutions of 1280×720 for the RGB image and 1024×768 for the depth image, respectively. The RGB images and depth images are aligned with the same frame rate (30 fps).

For *GF* tracking, we employed PlaneRCNN [59] for plane detection and a state-of-art line segmentation algorithm [28] for edge detections. Specifically, the plane detection algorithm takes the RGB images as input and is able to segment plane regions even for small objects. Combining the 2D plane detection results and depth information, our system can project the 2D plane position back to 3D so that users can view the final results in AR view. The plane detection showed state-of-art performance: the average precision (AP) reaches 0.365 with intersection over union (IoU) threshold 0.5 and depth error threshold of 0.3 m. For the edge detection, we adopt a learning-based line segmentation detection (LSD). The LSD is lightweight enough to run on mobile devices (48.6 fps on phones) while maintaining competitive performance (line matching average precision of 64.2). For both plane detection and edge detection algorithms, we adopt the same parameter settings as the original works and both algorithms can run in real-time at 30 fps.

The systems were developed with Unity3D (2019.4.16f1) and the user interfaces were supported by Microsoft Mixed Reality Toolkit (MRTK)[1]. The upper body joints were estimated from three-point tracking (head and two heads) via inverse kinematics (IK) [22]. We used the Python interface on the backpack computer mentioned above to run the optimization solver Gurobi [31]. The data on HoloLens and the backpack computer were transmitted through Transmission Control Protocol (TCP) [70] in the local network.

5 Application Scenarios

With AdapTUI, users are able to utilize nearby *GFs* in the scene to create tangible AR scenarios that can be deployed in different environments. Here we demonstrate several example scenarios to illustrate the versatility of AdapTUI.

5.1 Portable Car Play

Users are able to play with TUIs in different car positions with AdapTUI. As shown in Figure 5a, several TUIs including a digital map control pad, a plane-based speedometer display, an edge-based personal music player, and a car speaker volume controller linked to an edge of the door's armrest are initialized in the passenger seat of an SUV. Then when the user moves to the back seat in Figure 5b, the TUI mappings are adapted to the new environment. In the new environment, the digital map and the speedometer display are aligned with the headrest plane, and the music player is connected to the edge of the seat. Moreover, the car volume controller is now connected to the armrest of the backseat.

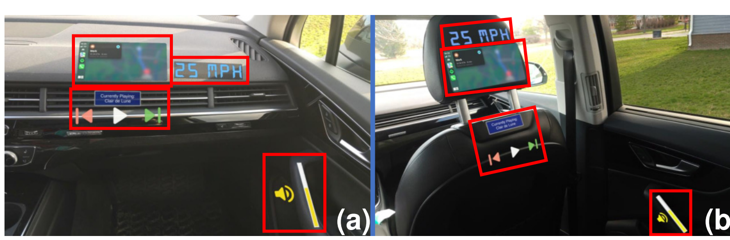

Fig. 5. The application scenarios of portable car play include 4 TUIs: 1) a plane-based digital map, 2) a plane-based speedometer 3) a music player, and 4) an edge-based car speaker volume controller. The user initializes the TUI mappings in the front seat (a) and uses AdapTUI to adapt the TUI mappings when the user moves to the backseat (b).

5.2 Efficient AR Workstation

AdapTUI can facilitate work efficiency for users who need to work in different places (e.g. cleaner, plumbers). For example, a plumber is shipped with several work-related TUIs (Figure 6): an instruction note that displays the current steps of fixing the plumb, an instruction controller to check the following/previous steps, a timer to record the work completion time, an AR to-do list that displays the planning of the day, and a photo button for taking photos for task completion. The plumber first initializes the TUI mappings in a bathroom (Figure 6a): 1) the instruction note attached to the cabinet's bottom edge, 2) the shutter button attached to the cabinet's vertical edge, 3) the timer

[1]https://github.com/microsoft/MixedRealityToolkit-Unity

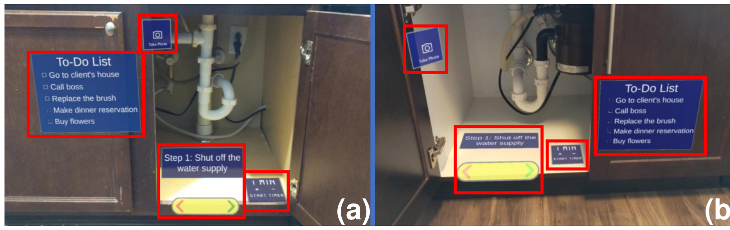

Fig. 6. The efficient AR workstation consists of 1) an edge-based shutter button, 2) an edge-based instruction, 3) a plane-based timer, and 4) a plane-based to-do list notepad.

attached to the bottom surface, and 4) the to-do list attached to the cabinet door. Then he goes to another home and uses AdapTUI for TUI adaptation in the new environment. Now the to-do list moves to the right-side cabinet surface, and the others remain similar to the initializations.

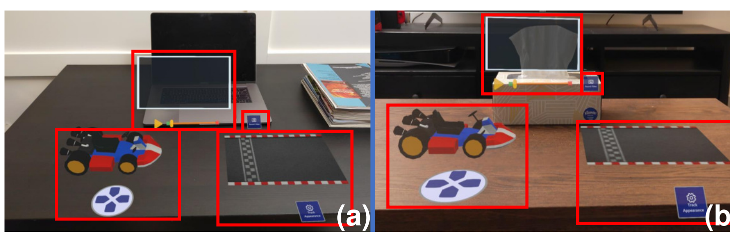

Fig. 7. The TUIs involved in the game scenario include 1) a plane-based car controller, 2) a plane-based button for changing track shape, 3) an edge-based button for recording AR games, and 4) an edge-based video player for reviewing the recordings and game trailers. A user first initializes the TUIs on a computer desk and then enjoys AdapTUI's adaptation results for the TUI mapping in the living room.

5.3 Entertainment/Game

We enable end-users to deploy TUI-based AR games in different scenarios. For instance, a user can play an AR car game by controlling indoor *GFs*. Specifically, a user can control the car's movement using a plane-attached panel, and change the track type through an edge-based TUI. The user can also record the game experience by pressing the edge-based recording button, and review the past recordings using a video player which is connected to a table surface (Figure 7a). When the user shifts to the new environment, our system adapts the TUI placement automatically (Figure 7b). The car controller and tracker pads are now mapped to the new table surface. Moreover, the video player is connected to the top edge of the tissue box.

6 System Evaluation

We conducted a user study to evaluate the effectiveness as well as generalizability of the optimization approach, and investigated overall system usability. We recruited 12 users (10 males and 2 females, ages ranging from 23 to 30). 11 out of 12 had used AR or VR applications on smartphones, tablets, or head-mounted devices while the rest user had heard about AR and VR. In this user study, we compared AdapTUI with a baseline method that automatically aligns the tangible input to the users' nearest *GF* without considering the sub-objectives (*GF association agreement, human factors, etc.*), similar to [9, 58]. None of the users had used any method of adaptation before. To avoid biased

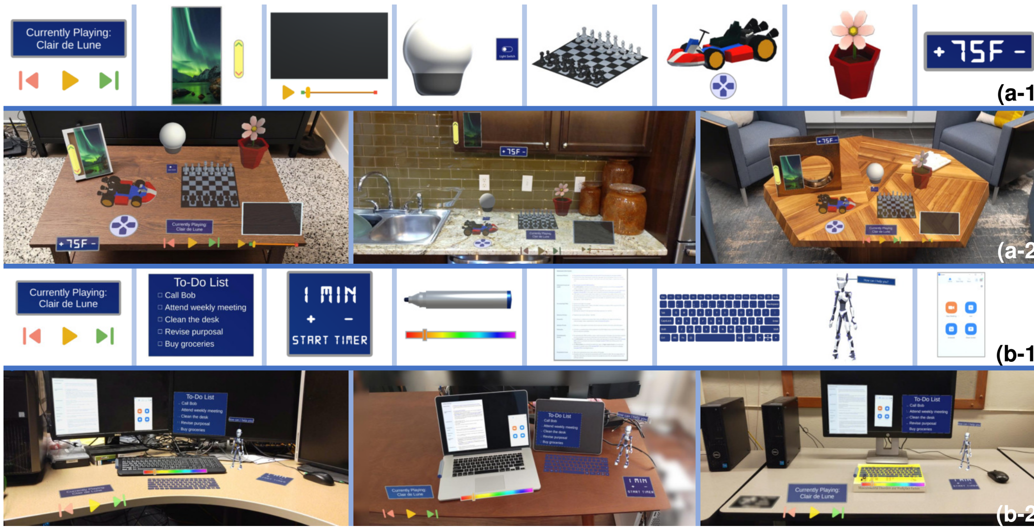

Fig. 8. User Study Setup. (a-1) The *home* TUIs: a music player, a virtual photo album, a video player, a light bulb, an AR chessboard, an AR car, an AR plant, and a thermostat. (b-1) The *office* TUIs: a music player, a to-do list, a timer, a color panel for an AR pen, an AR document, a digital keyboard, an AR agent, and a virtual conference interface. During adaptations for *home* TUIs (a-2) and *office* TUIs (b-2), the users initialized TUIs in a room of *prototyped scene* (left), and experienced adaptation results using two methods in the other two rooms (middle and right).

feedback, we did not inform users of which method we targeted to evaluate and shuffled the order of using the two methods. The whole study took around 1.5 hours and each user was paid with a 20 e-gift card. After a user came, we first introduced the TUI adaptation background and let users understand the workflow for both methods. Then the users could experience the HoloLens2 built-in tutorials before the user study officially starts. After finishing the adaptation process with the two methods, users completed a 5-scale Likert-type questionnaire and the standard System Usability Scale (SUS) questionnaire. Finally, we conducted a conversation-style interview with the users to obtain their subjective feedback about the two adaptation methods.

6.1 Study Design

TUIs. During the user study, we provided two sets of TUIs, which are divided and named based on their functions: *home* (Figure 8a-1) and *office* (Figure 8b-1), for users to perform the adaptation. For each set of TUIs, we provided eight TUIs for users to interact with. The *home* TUIs include a music player, a virtual photo album, a video player, a light bulb, an AR chessboard, an AR car, an AR plant, and a thermostat. And the *office* TUIs include virtual objects including a music player, a to-do list, a timer, a color panel for an AR pen, an AR document, a digital keyboard, an AR agent, and a virtual conference interface.

Environments. To test whether our adaptation method is effective in environments with distinct layouts, we chose three different rooms with distinct layouts for each set of TUIs. Specifically, we chose a living room, a kitchen, and a university's common space for *home* TUIs (Figure 8a-2), and an office, a dining room, and a lab for *office* TUIs (Figure 8b-2). The rooms were chosen based on the following attributes: 1) ubiquitous GFs are contained, and 2) the rooms vary in numbers, positions, and orientations of *GFs* (planes and edges). For each set of TUIs, one room was used as *prototyped* *scene* and the other rooms were used as *output scenes* for adaptation use. To avoid biased results, we divided users into three subgroups and each subgroup used a different room as *prototyped scene*. In addition, the order of the using two methods is shuffled. In total, we collected 2 × 8 (objects) × 2 (rooms) = 32 adaptation trials for each user's experience.

Methods. We compared AdapTUI with a baseline adaptation method. The baseline method adapts the TUI positions by selecting the nearest *GF* to users without taking the sub-objectives mentioned in Sec 3.3. We chose this baseline for several reasons. First, instead of focusing on individual sub-objectives, we aim to evaluate the overall effectiveness of AdapTUI. Drawing inspiration by prior works [9, 33, 58], which systematically evaluate the optimization algorithm and commonly employ basic adaptation methods without sub-objectives, such as manual adjustment or positioning UI elements at users' same relative positions, as baselines for comparisons, we select a basic automatic adaptation method. Here, the decision to opt for automatic adaptation over manual adjustments was motivated by avoiding time-consuming and naive manual placements. Further, as discussed in Sec 2.2, current optimization-based adaptation works cannot be directly applied to *GF* adaptations. Based on the above considerations, automatically selecting users' nearest *GF* seems the most practical alternative for comparison.

6.2 Procedure

During the adaptation evaluation for each set of TUIs, users were first asked to manually set the layouts for the *input scene* as described in Section 4.1 and then moved to the other two rooms to conduct the adaptation process. In the *output scene*, users checked the layouts generated by AdapTUI and the baseline method respectively. For layouts generated by each method, users were asked to adjust the layouts manually until they were satisfied. For each adaptation method, we compared the adaptation-generated results with the final manually adjusted results and recorded the number and time taken for manual adjustment.

6.3 Results and Discussions

All 12 users successfully completed the authoring and adaptation process. The quantitative results are shown in Figure 9, and the collected Likert-type results are shown in Figure 10.

6.3.1 Quantitative Results. Overall, users on average needed to perform 9.06 adjustments (SD = 2.15) for all four adaptation rooms using AdapTUI (Figure 9a). The average time taken was 5.29 minutes (SD = 1.39) for all the manual adjustments. On the other hand, the baseline method required users to conduct 16.92 adjustments (SD = 3.18) with an average time of 10.13 minutes (SD = 1.95) for TUI mappings. The results indicate AdapTUI could significantly reduce the time required for TUI mapping by 48%, and users made fewer manual adjustments to reach optimal mappings, with a 44% reduction. In addition, to investigate the effect of adaptation methods, we performed a one-way ANOVA test on the number and time of manual interaction adjustment results after data passed Kolomogorov-Smirnov normality test. The ANOVA results show that the adaptation methods have a significant impact on the adaptation methods ($F(1, 22) = 12.952$, $p < .001$), and the total time taken for manual adjustment ($F(1, 22) = 10.049$, $p < .001$).

6.3.2 Qualitative Results. To analyze the difference between the two methods, we performed the Wilcoxon signed-rank tests for the questionnaire results. In general, users acknowledged the convenience of using *GF*-based TUI in daily usage and showed a high preference for utilizing our optimization-based method (Q1, $AVG = 4.67$, $SD = 0.49$), $Z = -2.97$, $p = .03$. *"The optimization-based method tailors to my using habits when I need to touch the TUI. Compared with the other adaptation method, it put items in my nearby area and the TUIs were placed in the places I expected. (P2)"* Users first recognized that the automatic adaptation method is necessary for TUI mappings

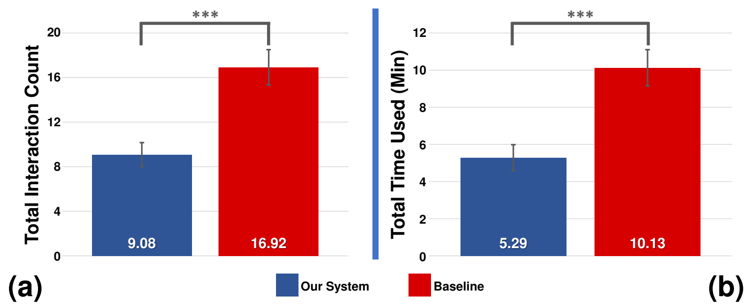

Fig. 9. Quantitative Results. (a) Total number of manual adjustments after adaptation in the four environments. (b) Total time taken for manual adjustments (***: p< .001).

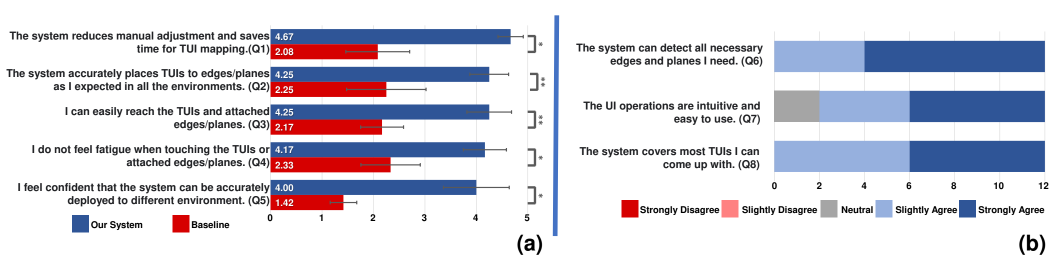

Fig. 10. Likert-type questionnaire results. (a) Comparison results between AdapTUI and the baseline method (*: p < 0.5, **: p< .01, ***: p< .001). (b) The overall system evaluation results.

and the versatility of the adaptation process (Q5, $AVG = 4.00$, $SD = 1.27$), $Z = -2.96$, $p = .03$. *"I can think of multiple scenarios that the adaptation system can be deployed to. For example, I placed TUI mappings for AR Games at my home and then I go to my friends' home and enjoy the same settings there. (P8)"* Users agreed that AdapTUI precisely select the *GF* for TUI mapping (Q2, $AVG = 4.25$, $SD = 0.75$) $Z = -2.584$, $p = .01$. *"With so many edges and planes in the environment, it's amazing that the TUI are mapped to edges and planes I anticipate them to be. (P10)"* From the comments, users were satisfied with the adaptation results generated by AdapTUI and agreed that the adaptation results fit their use preferences (Q3, $AVG = 4.25$, $SD = 0.86$), $Z = -2.88$, $p = .004$. *"I think the optimization-based method satisfies all my needs and I do not need to make further adjustments. (P5)"* In particular, users highly appreciated that AdapTUI takes ergonomics into consideration (Q4, $AVG = 4.17$, $SD = 0.83$), $Z = -2.23$, $p = .026$. *"For the office scenario, I imagined that I would sit in the chair during work and therefore placed TUIs in a relatively low position. I feel it is pretty convenient to use the optimization-based method because I can easily reach the TUIs. However, the automatic (baseline) adaptation places the TUIs at a high position so I have to stand up to reach the TUI. (P1)"* Although most of the users thought AdapTUI can satisfy their needs for adaptation, one user raised that *"it would be better if the optimization-method can adapt to the object materials I touched. (P3)"* We will discuss this concern later in Section 7. Users also acknowledged the coverage of AR contents and provided (Q8, $AVG = 4.5$, SD= 0.52). *"The system covers almost all TUI applications that I can come up with. It covers buttons, sliders, 2D interfaces and 3D AR objects. (P9)"* We also received positive feedback regarding the edge/plane detection (Q7, AVG= 4.33, SD= 0.78). *"It's straightforward that all the planes and edges are shown in the AR view. The necessary edges and planes for attaching the TUI are displayed by the system. (P12)"* Additionally, users complimented the UIs and the intuitive operations during authoring and adaptations (Q6, AVG = 4.67, SD = 0.49). *"It's easy to align the TUIs with the edges/planes with the auto-snapping functions. (P11)"* The standard SUS survey result of AdapTUI is 87.94 out of 100 with a standard deviation of 8.91 (Baseline: AVG=77.5, SD=7.81), showing the high usability of the system.

7 Limitations

While our optimization model demonstrated its effectiveness and generalization potential in the evaluation, we acknowledge that some limitations might affect the system's overall performance and user experiences. In the following, we categorize limitations into technical challenges and user-experience conflicts, and discuss alternative solutions to these challenges.

7.1 Technical Limitations

Curve and corner detections. Technically, our system chooses the most common geometric features, i.e., the straight edges and planes, as tangible input. While most users reflected satisfaction in the *GF* detection and TUI usage, some users raised that the corner and curve extraction can be added. *"The edge segmentation method extracts two straight edges rather than a continuous curve for the rounded corner. (P4)"* Similarly, another user came up with the idea of utilizing knob-like items as TUIs. *"I am wondering if I were a worker in a factory, can I interact with buttons on a workstation? (P6)"* Currently, as we focus on planes and straight edges, AdapTUI is not able to render TUI mappings for curved surfaces or curves. Extending the current straight edge detection to curve detection or corner detection [1, 44] might be a solution. Specifically, the corner/curve detections usually rely on learning-based methods to segment 3D point clouds and group curves [44, 83]. By applying curve detection and integrating curves into TUIs, we can enable users to interact with physical objects with more complex shapes, e.g. curved chair arms. Further, users' interaction gestures can also be naturally extended due to the increased complexity of object structures. For example, when a user interacts with a knob, the gesture of turning/twisting can be naturally considered as an interaction gesture. However, these learning-based curve/corner detection algorithms require demanding computational resources (4.1 G GPU memory for a single object) due to the complexity and large quantity of curves in the physical environment. Therefore, it is meaningful to explore rendering adequate curves in the physical environment while keeping real-time interactive experience for TUIs as future work.

7.2 User Experience Limitations

Opimization results conflicted with users desires. The optimization adaptation supported by AdapTUI focuses on maintaining spatial and ergonomic affordances when users are in different environments. At the same time, the optimization's generalizability was proved to be feasible in the user study. Yet, we found that the constraints are not always feasible. For example, AR contents aligned with a large-size *GF* in the *prototyped scene* may not be rendered to a small area in the *output scene* due to the size constraints (Sec 3.4), which may conflict with users' desire. Similarly, two TUI elements previously placed on a large area, e.g., a dining table, may not be rendered at the same time if users move to a spatially constrained context, like an airplane seat. In this case, our method chooses the TUI with a compatible size, which however may not be the user's preference. One possible solution is to record users' usage frequency for each element and the system optimizes the visibility of each element based on usage frequencies [58, 60]. However, this method requires long-term data collection to get stable adaptation results. Another potential solution can be adding customized constraints for optimization [66], which allows users to have more control over the optimization schemes. While this allows users to define their own constraints, it burdens users with more manual actions required compared with automatic adaptations.

Dynamic adaptation frequency. As mentioned in Section 4.1, our optimization method conducts the adaptation when there are substantial changes in users' status and nearby environment while prioritizing existing optimization results. However, we observed that users have different preferences for adaptation frequency. Some users preferred *"the TUIs reside with me wherever I am. For example, during a meeting, I want to sit on one side of a desk and interact with TUIs. After a while, I change sits with others and sit on the other side of the table but I still want the TUI to follow me. (P5)"*. Others like manually setting the pace of optimization: *"I only need one adaptation when I am in the kitchen and I do not need to find the TUIs all the time. (P12)"* Adding context-aware information like task performed and user's mental load could be a solution[23, 58]. Alternatively, we can apply learning-based methods or large language models (LLM) to predict users' preferences [6, 88]. Recently, language models like [81, 92] can turn users' actions from input photos into context and apply zero-shot learning to predict users' possible actions. We can use a similar approach to interpret users' interactions with AR environment and predict users' adaptation frequency.

Customization option of preferred/undesired areas. AdapTUI aims to provide users with an automatic adaptation process and the least manual prototyping steps. Yet, some users claimed that they hoped more customization options could be provided especially for the preferred or undesired interaction area. *"TUIs might not be suitable to be placed in dangerous or hazardous areas, like an oven in the kitchen and some dangerous places in a factory. Similarly, I may only need a specific area for TUI adaptation instead of the whole room. (P11)"* SemanticAdapt [9] addresses the undesired area problem by letting users manually set the objects to avoid. Another possible solution is to predict user preferences using reinforcement learning [5, 27, 29, 58, 77]. Reinforcement learning-based methods enable systems to adapt to individual user's preferences over time and reduce the need for manual input. Using reinforcement learning, our system can first initialize a generic TUI adaptation for all users. Then, as users make adjustments to the generic adaptation, our system records the users' modifications (i.e. moving TUIs from generic adaptation locations to preferred locations) and tailors the optimization process (weighting of individual sub-objectives) for each individual user.

8 Discussion & Future Work

Add object materials as adaptation factors. AdapTUI addresses using *GF*-association and human factors as main optimization factors while taking visibility and spatio-temporal agreement into consideration. Although most users praised the optimization design of AdapTUI, a user came up with the idea that TUI interaction feeling is an important factor to explore. *"The feeling of touching the soft cloth of the sofa is pretty different from touching the wooden desk. It would be better if the optimization model could optimize according to touch feelings. (P4)"* Previous works have provided insight into this problem and provide detection solutions for touches with different textures [68, 74]. In future work, it would improve the optimization method to add object materials and users' possible hand interaction with the TUI as optimization input. Further, as different object materials provide various haptic feedback, they can also be used for multisensory interactions in AR space (e.g., an AR shooting gun causes different shapes of bullet holes when bullets hit different materials) [7].

Add physiological data as potential input. AdapTUI targets TUI adaptations for keeping spatial and ergonomic affordances, and takes users' joints and *GF* as main inputs. Yet, our system can benefit more if physiological data/metrics can be taken into account. Specifically, physiological data, such as heart rate, skin conductance, or eye movements, provide informative clues of users' health, emotion, and cognition status [24, 53, 76]. This allows us to better monitor users' ergonomic status and optimize towards better conditions of users' health. For example, we can use eye-tracking measures to record eye blinks, and predict users' mental load.

Balance between potential privacy risks and adaptations. AdapTUI supports users to enter different locations and adapts the TUI placement in the new environment. In addition, AdapTUI monitors physical objects and other people in the environment to avoid occlusion (Sec 3.3.3). However, privacy may become a problem for shared areas or private places. For example, in a shared working space, colleagues might be unpleasant if they are captured by the camera. Studies on balancing privacy and effective adaptation could be one research direction of future adaptation models.

Combining adaptations for TUIs and mid-air UIs. Currently, our system is designed for *GF*-based TUIs while ensuring users' ergonomics. Technically, our system has the capability to be expanded to accommodate mid-air UI adaptations. On the other hand, it's worth noting that TUIs have inherent limitations, as their interactive spaces are confined to users' surroundings whereas mid-air UIs offer the flexibility to be positioned over the entire environment [42, 50]. To address diverse user needs in the environment, studies on integrating adaptations for both TUIs and mid-air UIs and how to balance between TUIs and mid-air UIs could be one of the research directions for future adaptation research.

9 Conclusion

In this work, we present AdapTUI, an adaptation system for automatically aligning geometric-based TUIs to *GFs* in different environments. AdapTUI enables users to first initialize their preferences for TUI mapping with the environment, and then perform automatic mapping when users move to new environments. We first discussed the benefits of automatically adapted mapping for TUI applications as well as the importance of consistent spatial and ergonomics affordances during the adaptation process. Then, we proposed an optimization model for the adaptation process, where adaptation objectives including consistent geometric association, human factors, visibility, and spatio-temporal consistency were designed. Guided by the adaptation objectives, we define four sub-objective functions for the optimization. Further, we demonstrated three application scenarios, namely Portable Car Play, Efficient AR Workstation, and Entertainment/Game, that exploit the capability of our system. To explore our system's generalizability and the effectiveness of the adaptation model, we compared our optimization model with a baseline method. The evaluation results showed that AdapTUI decreased the manual adjustment number by 44% and users preferred using AdapTUI. To sum up, we believe that AdapTUI exposes a promising step towards automatically and reasonably integrating geometric-based TUIs into users' ambient environment.

Acknowledgements

We thank the reviewers for their invaluable feedback. This work is partially supported by the NSF under the Future of Work at the Human Technology Frontier (FW-HTF) 1839971. We also acknowledge the Feddersen Distinguished Professorship Funds. Any opinions findings, and conclusions are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] Syeda Mariam Ahmed, Yan Zhi Tan, Chee Meng Chew, Abdullah Al Mamun, and Fook Seng Wong. 2018. Edge and corner detection for unorganized 3d point clouds with application to robotic welding. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7350–7355. http://arxiv.org/abs/1809.10468

[2] Vincent Becker, Sandro Kalbermatter, Simon Mayer, and Gábor Sörös. 2019. Tailored controls: Creating personalized tangible user interfaces from paper. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. 289–301. https://doi.org/10.1145/3461778.3462059

[3] Frank Beruscha, Katharina Mueller, and Thorsten Sohnke. 2020. Eliciting tangible and gestural user interactions with and on a cooking pan. In *Proceedings of Mensch und Computer 2020*. Association for Computing Machinery, New York, NY, USA, 399–408. https://doi.org/10.1145/3404983.3405516

[4] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Magic desk: bringing multi-touch surfaces into desktop work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2511–2520. https://doi.org/10.1145/1978942.1979309

[5] Micah Carroll, Dylan Hadfield-Menell, Stuart Russell, and Anca Dragan. 2021. Estimating and penalizing preference shift in recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 661–667. https://doi.org/10.1145/3460231.3478849
[6] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. 2024. MaxMin-RLHF: Towards Equitable Alignment of Large Language Models with Diverse Human Preferences. *arXiv preprint arXiv:2402.08925* (2024). https://arxiv.org/abs/2402.08925
[7] Long Chen, Karl Francis, and Wen Tang. 2017. [POSTER] Semantic Augmented Reality Environment with Material-Aware Physical Interactions. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. IEEE, 135–136. https://arxiv.org/abs/1708.01208
[8] Kai-Yin Cheng, Rong-Hao Liang, Bing-Yu Chen, Rung-Huei Laing, and Sy-Yen Kuo. 2010. iCon: utilizing everyday objects as additional, auxiliary and instant tabletop controllers. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1155–1164. https://doi.org/10.1145/1753326.1753499
[9] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. Semanticadapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 282–297. https://doi.org/10.1145/3472749.3474750
[10] Yi Fei Cheng, Christoph Gebhardt, and Christian Holz. 2023. InteractionAdapt: Interaction-driven Workspace Adaptation for Situated Virtual Reality Environments. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14. https://doi.org/10.1145/3586183.3606717
[11] Yi Fei Cheng, Tiffany Luong, Andreas Rene Fender, Paul Streli, and Christian Holz. 2022. ComforTable User Interfaces: Surfaces Reduce Input Error, Time, and Exertion for Tabletop and Mid-air User Interfaces. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 150–159. https://doi.org/10.1109/ISMAR55827.2022.00029
[12] Christian Corsten, Ignacio Avellino, Max Möllers, and Jan Borchers. 2013. Instant user interfaces: repurposing everyday objects as input devices. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. 71–80. https://doi.org/10.1145/2512349.2512799
[13] Christian Corsten, Chat Wacharamanotham, and Jan Borchers. 2013. Fillables: everyday vessels as tangible controllers with adjustable haptics. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2129–2138. https://doi.org/10.1145/2468356.2468732
[14] Dragoş Datcu, Stephan Lukosch, and Frances Brazier. 2015. On the usability and effectiveness of different interaction types in augmented reality. *International Journal of Human-Computer Interaction* 31, 3 (2015), 193–209. https://doi.org/10.1080/10447318.2014.994193
[15] Zhi-Chao Dong, Wenming Wu, Zenghao Xu, Qi Sun, Guanjie Yuan, Ligang Liu, and Xiao-Ming Fu. 2021. Tailored reality: Perception-aware scene restructuring for adaptive vr navigation. *ACM Transactions on Graphics (TOG)* 40, 5 (2021), 1–15. https://doi.org/10.1145/3470847
[16] Adam Drogemuller, James Walsh, Ross T Smith, Matt Adcock, and Bruce H Thomas. 2021. Turning everyday objects into passive tangible controllers. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 1–4. https://doi.org/10.1145/3430524.3442460
[17] Ruofei Du, Alex Olwal, Mathieu Le Goc, Shengzhi Wu, Danhang Tang, Yinda Zhang, Jun Zhang, David Joseph Tan, Federico Tombari, and David Kim. 2022. Opportunistic Interfaces for Augmented Reality: Transforming Everyday Objects into Tangible 6DoF Interfaces Using Ad hoc UI. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–4. https://doi.org/10.1145/3491101.3519911
[18] Andreas Dünser, Karin Steinbügl, Hannes Kaufmann, and Judith Glück. 2006. Virtual and augmented reality as spatial ability training tools. In *Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI*. 125–132. https://doi.org/10.1145/1152760.1152776
[19] João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. 2021. XRgonomics: facilitating the creation of ergonomic 3D interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3411764.3445349
[20] João Marcelo Evangelista Belo, Mathias N Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. 2022. AUIT–the Adaptive User Interfaces Toolkit for Designing XR Applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–16. https://doi.org/10.1145/3526113.3545651
[21] Andreas Fender, Philipp Herholz, Marc Alexa, and Jörg Müller. 2018. Optispace: Automated placement of interactive 3d projection mapping content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3173574.3173843
[22] FinalIK 2023. FinalIK. https://assetstore.unity.com/packages/tools/animation/final-ik-14290
[23] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–13. https://doi.org/10.1145/2816795.2818057

[24] Thomas Fritz, Andrew Begel, Sebastian C Müller, Serap Yigit-Elliott, and Manuela Züger. 2014. Using psycho-physiological measures to assess task difficulty in software development. In *Proceedings of the 36th international conference on software engineering*. 402–413. https://doi.org/10.1145/2568225.2568266

[25] Markus Funk, Oliver Korn, and Albrecht Schmidt. 2014. An augmented workplace for enabling user-defined tangibles. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 1285–1290. https://doi.org/10.1145/2559206.2581142

[26] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. 2014. FLARE: Fast layout for augmented reality applications. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE, 207–212. https://doi.org/10.1109/ISMAR.2014.6948429

[27] Chengqian Gao, Ke Xu, Kuangqi Zhou, Lanqing Li, Xueqian Wang, Bo Yuan, and Peilin Zhao. 2022. Value penalized q-learning for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2008–2012. https://doi.org/10.1145/3477495.3531796

[28] Geonmo Gu, Byungsoo Ko, SeoungHyun Go, Sung-Hyun Lee, Jingeun Lee, and Minchul Shin. 2022. Towards light-weight and real-time line segment detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 726–734. https://doi.org/10.1609/aaai.v36i1.19953

[29] Enrico Guerra, Benedikt Kurz, and Jonas Bräucker. 2022. An Extension to the Keystroke-Level Model for Extended Reality Interactions. https://doi.org/10.18420/muc2022-mci-src-443

[30] Aakar Gupta, Bo Rui Lin, Siyi Ji, Arjav Patel, and Daniel Vogel. 2020. Replicate and reuse: Tangible interaction design for digitally-augmented physical media objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3313831.3376139

[31] Gurobi 2023. Gurobi Optimizer. https://www.gurobi.com.

[32] Lei Han, Tian Zheng, Yinheng Zhu, Lan Xu, and Lu Fang. 2020. Live semantic 3d perception for immersive augmented reality. *IEEE transactions on visualization and computer graphics* 26, 5 (2020), 2012–2022. https://doi.org/10.1109/TVCG.2020.2973477

[33] Violet Yinuo Han, Hyunsung Cho, Kiyosu Maeda, Alexandra Ion, and David Lindlbauer. 2023. BlendMR: A Computational Method to Create Ambient Mixed Reality Interfaces. *Proceedings of the ACM on Human-Computer Interaction* 7, ISS (2023), 217–241. https://doi.org/10.1145/3626472

[34] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 441–450. https://doi.org/10.1145/2047196.2047255

[35] Fengming He, Xiyun Hu, Jingyu Shi, Xun Qian, Tianyi Wang, and Karthik Ramani. 2023. Ubi Edge: Authoring Edge-Based Opportunistic Tangible User Interfaces in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14. https://doi.org/10.1145/3544548.3580704

[36] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D puppetry: a kinect-based interface for 3D animation. In *UIST*, Vol. 12. Citeseer, 423–434. https://doi.org/10.1145/2380116.2380170

[37] Steven J Henderson and Steven Feiner. 2008. Opportunistic controls: leveraging natural affordances as tangible user interfaces for augmented reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. 211–218. https://doi.org/10.1145/1450579.1450625

[38] Jaylin Herskovitz, Yi Fei Cheng, Anhong Guo, Alanson P Sample, and Michael Nebeling. 2022. XSpace: An Augmented Reality Toolkit for Enabling Spatially-Aware Distributed Collaboration. *Proceedings of the ACM on Human-Computer Interaction* 6, ISS (2022), 277–302. https://doi.org/10.1145/3567721

[39] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1957–1967. https://doi.org/10.1145/2858036.2858134

[40] Valentin Heun, Shunichi Kasahara, and Pattie Maes. 2013. Smarter objects: using AR technology to program physical objects and their interactions. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 961–966. https://doi.org/10.1145/2468356.2468528

[41] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1063–1072. https://doi.org/10.1145/2556288.2557130

[42] Eva Hornecker and Jacob Buur. 2006. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 437–446. https://doi.org/10.1145/1124772.1124838

[43] HP Z VR Backpack Workstation 2022. HP Z VR Backpack Workstation: VR this powerful is no toy. https://www.hp.com/us-en/shop/mdp/hp-z-vr-backpack-g1-workstation--1

[44] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. 2020. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 222–239. https://doi.org/10.1007/978-3-030-58565-5_14

[45] Intel RealSense LiDAR Camera L515 2022. Intel® RealSense™ LiDAR Camera L515. https://www.intelrealsense.com/lidar-camera-l515/

[46] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 234–241. https://doi.org/10.1145/258549.258715

[47] Sungjune Jang, Lawrence H Kim, Kesler Tanner, Hiroshi Ishii, and Sean Follmer. 2016. Haptic edge display for mobile tactile interaction. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 3706–3716. https://doi.org/10.1145/2858036.2858264

[48] Nikhita Joshi and Daniel Vogel. 2019. An evaluation of touch input at the edge of a table. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3290605.3300476

[49] Annie Kelly, R Benjamin Shapiro, Jonathan de Halleux, and Thomas Ball. 2018. ARcadia: A rapid prototyping platform for real-time tangible interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8. https://doi.org/10.1145/3173574.3173983

[50] M Kim and Mary Lou Maher. 2005. Comparison of designers using a tangible user interface and a graphical user interface and the impact on spatial cognition. *Proc. Human Behaviour in Design* 5 (2005), 14.

[51] Yining Lang, Wei Liang, and Lap-Fai Yu. 2019. Virtual agent positioning driven by scene semantics in mixed reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 767–775. https://doi.org/10.1109/VR.2019.8798018

[52] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional.

[53] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.

[54] Gun A Lee, Claudia Nelles, Mark Billinghurst, and Gerard Jounghyun Kim. 2004. Immersive authoring of tangible augmented reality applications. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 172–181. https://doi.org/10.1109/ISMAR.2004.34

[55] Changyang Li, Wanwan Li, Haikun Huang, and Lap-Fai Yu. 2022. Interactive augmented reality storytelling guided by scene semantics. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–15. https://doi.org/10.1145/3528223.3530061

[56] Rong-Hao Liang, Kai-Yin Cheng, Liwei Chan, Chuan-Xhyuan Peng, Mike Y Chen, Rung-Huei Liang, De-Nian Yang, and Bing-Yu Chen. 2013. GaussBits: magnetic tangible bits for portable and occlusion-free near-surface interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1391–1400. https://doi.org/10.1145/2470654.2466185

[57] Wei Liang, Xinzhe Yu, Rawan Alghofaili, Yining Lang, and Lap-Fai Yu. 2021. Scene-aware behavior synthesis for virtual pets in mixed reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3411764.3445532

[58] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 147–160. https://doi.org/10.1145/3332165.3347945

[59] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. 2019. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4450–4459.

[60] Feiyu Lu and Doug A Bowman. 2021. Evaluating the potential of glanceable ar interfaces for authentic everyday uses. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 768–777. https://doi.org/10.1109/VR50410.2021.00104

[61] Jacob Boesen Madsen, Markus Tatzqern, Claus B Madsen, Dieter Schmalstieg, and Denis Kalkofen. 2016. Temporal coherence strategies for augmented reality labeling. *IEEE transactions on visualization and computer graphics* 22, 4 (2016), 1415–1423. https://doi.org/10.1109/TVCG.2016.2518318

[62] Lynn McAtamney and E Nigel Corlett. 1993. RULA: a survey method for the investigation of work-related upper limb disorders. *Applied ergonomics* 24, 2 (1993), 91–99. https://doi.org/10.1016/0003-6870(93)90080-S

[63] Roberto A Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 759–771. https://doi.org/10.1145/3126594.3126605

[64] Michael Nielsen, Moritz Störring, Thomas B Moeslund, and Erik Granum. 2004. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers 5*. Springer, 409–420.

[65] Takehiro Niikura, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2014. Anywhere surface touch: utilizing any surface as an input area. In *Proceedings of the 5th Augmented Human International Conference*. 1–8. https://doi.org/10.1145/2582051.2582090

[66] Aziz Niyazov, Barrett Ens, Kadek Ananta Satriadi, Nicolas Mellado, Loïc Barthe, Tim Dwyer, and Marcos Serrano. 2023. User-Driven Constraints for Layout Optimisation in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16. https://doi.org/10.1145/3544548.3580873
[67] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D Wilson. 2016. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1233–1244. https://doi.org/10.1145/2858036.2858250
[68] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 31–40. https://doi.org/10.1145/2501988.2501989
[69] James Patten, Hiroshi Ishii, Jim Hines, and Gian Pangaro. 2001. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 253–260. https://doi.org/10.1145/365024.365112
[70] Jon Postel. 1981. *Transmission control protocol*. Technical Report.
[71] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*. 1–18. https://doi.org/10.1145/3491102.3517665
[72] Jan Riemann, Martin Schmitz, Alexander Hendrich, and Max Mühlhäuser. 2018. FlowPut: Environment-Aware Interactivity for Tangible 3D Objects. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–23. https://doi.org/10.1145/3191763
[73] Scott D Roth. 1982. Ray casting for modeling solids. *Computer graphics and image processing* 18, 2 (1982), 109–144. https://doi.org/10.1016/0146-664X(82)90169-1
[74] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 483–492. https://doi.org/10.1145/2207676.2207743
[75] Kadek Ananta Satriadi, Jim Smiley, Barrett Ens, Maxime Cordeil, Tobias Czauderna, Benjamin Lee, Ying Yang, Tim Dwyer, and Bernhard Jenny. 2022. Tangible globes for data visualisation in augmented reality. In *CHI Conference on Human Factors in Computing Systems*. 1–16. https://doi.org/10.1145/3491102.3517715
[76] Ekaterina R Stepanova, John Desnoyers-Stewart, Alexandra Kitson, Bernhard E Riecke, Alissa N Antle, Abdallah El Ali, Jeremy Frey, Vasiliki Tsaknaki, and Noura Howell. 2023. Designing with Biosignals: Challenges, Opportunities, and Future Directions for Integrating Physiological Signals in Human-Computer Interaction. In *Companion Publication of the 2023 ACM Designing Interactive Systems Conference*. 101–103. https://doi.org/10.1145/3563703.3591454
[77] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 214–221. https://doi.org/10.1609/aaai.v34i01.5353
[78] Tomu Tahara, Takashi Seno, Gaku Narita, and Tomoya Ishikawa. 2020. Retargetable AR: Context-aware augmented reality in indoor scenes based on 3D scene graph. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 249–255. https://doi.org/10.1109/ISMAR-Adjunct51615.2020.00072
[79] Kashyap Todi, Donald Degraen, Brent Berghmans, Axel Faes, Matthijs Kaminski, and Kris Luyten. 2016. Purpose-centric appropriation of everyday objects as game controllers. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2744–2750. https://doi.org/10.1145/2851581.2892448
[80] Deepak Tolani and Norman I Badler. 1996. Real-time inverse kinematics of the human arm. *Presence: Teleoperators & Virtual Environments* 5, 4 (1996), 393–401. https://doi.org/10.1162/pres.1996.5.4.393
[81] Tiffany Tseng, Ruijia Cheng, and Jeffrey Nichols. 2024. Keyframer: Empowering Animation Design using Large Language Models. *arXiv preprint arXiv:2402.06071* (2024). https://arxiv.org/abs/2402.06071
[82] Karl DD Willis, Takaaki Shiratori, and Moshe Mahler. 2013. HideOut: mobile projector interaction with tangible objects and surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. 331–338. https://doi.org/10.1145/2460625.2460682
[83] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. 2021. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 915–924.
[84] Robert Xiao, Chris Harrison, and Scott E Hudson. 2013. WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 879–888. https://doi.org/10.1145/2470654.2466113
[85] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. Direct: Making touch tracking on ordinary surfaces practical with hybrid depth-infrared sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. 85–94. https://doi.org/10.1145/2992154.2992173
[86] Robert Xiao, Scott Hudson, and Chris Harrison. 2017. Supporting responsive cohabitation between virtual interfaces and physical objects on everyday surfaces. *Proceedings of the ACM on Human-Computer Interaction* 1, EICS (2017),

1–17. https://doi.org/10.1145/3095814

[87] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D Wilson, and Hrvoje Benko. 2018. MRTouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1653–1660. https://doi.org/10.1109/TVCG.2018.2794222

[88] Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. 2023. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 3046–3056. https://doi.org/10.1145/3580305.3599400

[89] Zi-Ming Ye, Jun-Long Chen, Miao Wang, and Yong-Liang Yang. 2021. PAVAL: Position-Aware Virtual Agent Locomotion for Assisted Virtual Reality Navigation. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 239–247. https://doi.org/10.1109/ISMAR52148.2021.00039

[90] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86* 30, 4 (2011). https://doi.org/10.1145/2010324.1964981

[91] Yang Zhang, Chouchang Yang, Scott E Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++ room-scale interactive and context-aware sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15. https://doi.org/10.1145/3173574.3173847

[92] Haoyu Zheng, Wenqiao Zhang, Yaoke Wang, Hao Zhou, Jiang Liu, Juncheng Li, Zheqi Lv, Siliang Tang, and Yueting Zhuang. 2024. LASER: Tuning-Free LLM-Driven Attention Control for Efficient Text-conditioned Image-to-Animation. https://arxiv.org/abs/2404.13558

[93] Qian Zhou, Sarah Sykes, Sidney Fels, and Kenrick Kin. 2020. Gripmarks: Using Hand Grips to Transform In-Hand Objects into Mixed Reality Input. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3313831.3376313

[94] Zhengzhe Zhu, Ziyi Liu, Tianyi Wang, Youyou Zhang, Xun Qian, Pashin Farsak Raja, Ana Villanueva, and Karthik Ramani. 2022. MechARspace: An Authoring System Enabling Bidirectional Binding of Augmented Reality with Toys in Real-time. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–16. https://doi.org/10.1145/3526113.3545668

Received 2024-02-22; accepted 2024-05-30