SemanticAdapt: Optimization-based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections

Yifei Cheng*
Department of Computer Science and Technology, Tsinghua University
Swarthmore College

Yukang Yan†
Department of Computer Science and Technology, Tsinghua University

Xin Yi
Institute for Network Sciences and Cyberspace, Tsinghua University

Yuanchun Shi
Department of Computer Science and Technology, Tsinghua University

David Lindlbauer
Human-Computer Interaction Institute, Carnegie Mellon University

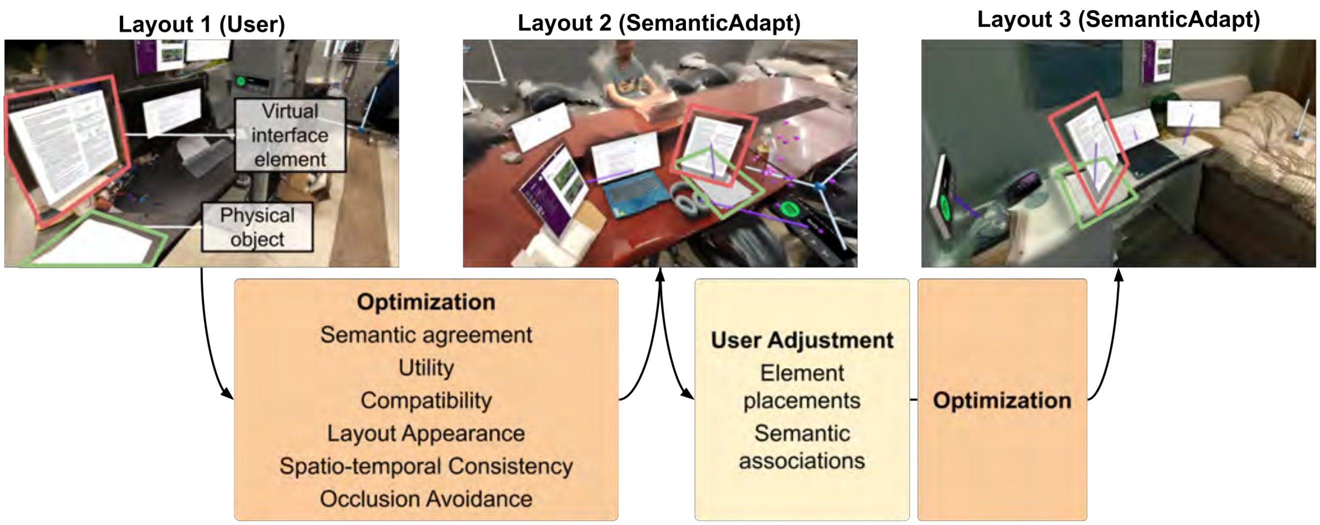

Figure 1: We propose an optimization-based approach to automatically adapt MR layouts between different environments. Here, SemanticAdapt adapts the user-created layout from an office (*layout 1*) to a conference room (*layout 2*), and then to a bedroom (*layout 3*). Semantic associations (*purple*) between virtual interface elements and physical objects are considered for the adaptation. SemanticAdapt places the PDF reader (*red*) consistently in close proximity with physical objects (*green*) related to paper or reading for different environments. Users may optionally adjust element placements and semantic connections. These changes are taken into account in the following adaptations.

ABSTRACT

We present an optimization-based approach that automatically adapts Mixed Reality (MR) interfaces to different physical environments. Current MR layouts, including the position and scale of virtual interface elements, need to be manually adapted by users whenever they move between environments, and whenever they switch tasks. This process is tedious and time consuming, and arguably needs to be automated for MR systems to be beneficial for end users. We contribute an approach that formulates this challenge as a combinatorial optimization problem and automatically decides the placement of virtual interface elements in new environments. To achieve this, we exploit the semantic association between the virtual interface elements and physical objects in an environment. Our optimization furthermore considers the utility of elements for users' current task, layout factors, and spatio-temporal consistency to previous layouts. All those factors are combined in a single linear program, which is used to adapt the layout of MR interfaces in real time. We demonstrate a set of application scenarios, showcasing the versatility and applicability of our approach. Finally, we show that compared to a naive adaptive baseline approach that does not

*This work was done while Yifei Cheng was an intern at Tsinghua University.
†The corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST'21, October 2021, Virtual
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474750

take semantic associations into account, our approach decreased the number of manual interface adaptations by 33%.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**; **Virtual reality**; *User interface management systems.*

KEYWORDS

Mixed Reality, Computational interaction, Adaptive user interfaces

ACM Reference Format:
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 (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3472749.3474750

1 INTRODUCTION

Mixed Reality (MR) interfaces hold the promise to provide efficient and flexible access to digital information, embedded in the physical world. The ability to arrange virtual interface elements freely in space has the potential to facilitate both productivity and casual interactions, and enable personalized digital spaces beyond what is possible with current desktop computers.

Consider the following scenario: a researcher surveys papers in their office wearing a MR headset. Virtual interface elements such as documents, figures, videos and 3D data visualizations, as well as shortcut icons to commodity applications (e. g. calendar) are arranged to form a workspace. The workspace integrates well into their current physical environment, i. e. an office with a desk, laptop, photos, books, decoration and a large coffee mug. The researcher has carefully laid out the items based on their own preference to achieve high efficiency and comfort. Moving to a different environment (e. g. a coffee shop) to work on the same task, however, breaks this layout. The researcher has to manually re-arrange the layout to fit the new physical environment. This process is time consuming and tedious, particularly considering that this adaptation needs to occur every time the researcher changes their environment. We argue that for any MR interface to be useful, it must adapt automatically to such changes.

Current research tackles this challenge through approaches that adapt MR layouts based on user behavior [14], visibility and occlusion of virtual interface elements (e. g. [2, 3, 20]) or geometric features of the environment [18]. Other approaches take users' state such as cognitive load [32] or gaze patterns [19] into account. None of these approaches, however, considers a tight semantic connection between the virtual interface elements and the physical world for MR layout adaptation. A virtual calendar application, for example, can always be anchored to time-related physical objects (e. g. a clock). We argue that such *semantic associations* are a key factor to keeping MR layouts *consistent* across multiple environments, enabling users to exploit their spatial memory and to increase the predictability of MR layouts. We aim at incorporating this consistency for the successful automatic adaptation of MR interfaces.

We contribute an novel approach to automatically adapt MR layouts to users' environments, called SemanticAdapt. Our optimization-based approach first considers a set of factors for individual element placements: 1) the semantic associations between virtual interface elements and physical objects, 2) utility of the virtual interface elements with respect to a task, and 3) compatibility between virtual elements and the spatial properties of potential placement locations. We additionally optimize for the overall layout arrangement including 4) layout appearance, 5) spatio-temporal consistency, and 6) occlusion avoidance. We contribute a novel formulation that solves the problem of interface adaptation as a single constrained optimization problem, in contrast to previous approaches that relied on the combination of multiple methods (e. g. [32]).

The inputs for our approach are a specification of the current environment, including usable space and existing physical objects, and the user's current task and its accompanying virtual interface elements. We compute the semantic connections between physical objects and virtual interface elements using a lexical database (WordNet [35]). Our approach additionally considers historic data of previous layouts in different environments. Based on the input, our approach decides on the placement of the virtual elements using integer programming. We allow users to fine-tune the layout, and alter semantic connections explicitly, which is leveraged as objectives and constraints in the optimization. Our approach runs at interactive rates, and we envision it to be executed whenever users change their environment or task.

To inform the design of our optimization-based approach, we first conducted a formative study. Twelve participants were asked to manually design MR layouts for different tasks and environments. Results indicate the importance of semantic connections, and revealed a set of layout behaviors, which are taken into account by our approach. We then detail our approach, including a constrained optimization problem formulation. To evaluate the efficacy of our approach, we compare our approach with a baseline method. The baseline adapts a MR layout to fit the usable space of a new environment, but does not consider semantic connections. Results indicate that our approach requires 33 % fewer manual adaptations by users and the generated layouts were perceived to be significantly more suitable for the physical environments. Finally, we detail a set of scenarios, showcasing the applicability of our approach. We release the SemanticAdapt source code for academic research.[1]

In summary, we make the following contributions.

- We present SemanticAdapt, a real-time optimization-based MR layout adaptation algorithm that leverages the semantic associations between virtual interface elements and physical objects.
- We explore user considerations and strategies for designing and adapting MR layouts in different environments through a formative user study, informing the design of SemanticAdapt and other automatic adaptation methods.
- We compare SemanticAdapt with a baseline method in a layout adaptation task in two typical usage scenarios in different environments. Results show that our approach decreases the number of manual adaptations by 33 % and was preferred by users.

[1] https://github.com/ycheng14799/SemanticAdapt

2 RELATED WORK

2.1 Adaptive Mixed Reality Interface

In contrast to screen-based devices, where interface elements are bound to self-contained 2D surfaces, MR interfaces are unconstrained, often embedded into users' environments, and therefore inherently context sensitive (cf. Lindlbauer et al. [32]). While the unrestricted nature of MR interfaces is powerful, it is difficult to determine how the interface should be displayed in changing usage contexts (cf. Azuma et al. [1]). A variety of works approached this challenge by casting it as a view management and label placement problem. Azuma and Furmanski [2], for example, contributed labeling placement strategies based on various clustering techniques. This was later extended by Gebhardt et al. [19], who adapted the visibility of virtual labels by employing reinforcement learning. Julier et al. [28] used filtering to selectively display information to avoid overwhelming users. Bell et al. [3] proposed to track real entities to prevent occlusions between interface elements and the environment. In this context, DiVerdi et al. [9] proposed the concept of level of detail interfaces, which present different amounts of information to users. All these works are concerned with visibility and occlusion, whereas our work is concerned with the connection between the virtual interface elements and the physical environments.

More recent works proposed taking additional information of the environment geometry and user state into consideration. For instance, Grasset et al. [20] presented an image-based approach that takes into consideration the presence of edges and the visually salient regions of the environment. Gal et al.'s system [18] decided on virtual content placements based on the scene geometry. Tahara et al. [49] defined a scene graph to record the spatial relationship between the virtual content and the physical objects, and ensured it is consistent when the user moved to other environments. Fender et al. leveraged recordings of user behavior to determine the optimal position of displays (HeatSpace [15]) and projected elements (OptiSpace [14]). Grubert et al. [21] argued for the importance of context-awareness of AR in an industrial application context. The approach by Lindlbauer et al. [32] decided on the type of placement (world- or view- anchored) and level of detail of virtual content using a mix of rule-based decision making and combinatorial optimization, distinctly taking into consideration the user's cognitive load.

In our work, we similarly are concerned with the display of virtual interface elements in a physical environment. We work on the specific problem of retargeting world-embedded MR layouts from one environment to another. Our approach adapts the position and orientation of virtual elements within new environments. We distinguish our work with our focus on semantic associations between virtual interface elements and environment objects. Factors such as cognitive load estimation [32] or user behavior [14, 15] could readily be integrated in our optimization scheme. In this work, however, we focus on highlighting and exploring the benefits of considering semantic associations for adapting MR layouts.

2.2 Optimization-based UI design

This work is based on prior research on automating UI generation. We apply established methods, specifically combinatorial optimization (see e. g. Oulasvirta et al. [39] and Feit [13] for overviews), but contribute a formulation that is unique to adding virtual-physical connections to MR layout generation. Gajos et al. [17], for example, used branch-and-bound optimization to generate interface renditions that meet device constraints and minimized user effort. Karrenbauer and Oulasvirta [29] used integer programming to optimize keyboard layouts. Dayama et al. [8], O'Donovan et al. [38], and Todi et al. [50] leveraged optimization as a means to support users in the design process. Park et al. [40] presented a method to generate distributed user interfaces based on factors like device characteristics and user roles. Our work uses the technique of integer programming for MR layout adaptation based on several considerations such as the utility of interface elements drawn from this body of work. Different from previous approaches such as Lindlbauer et al. [32] that require a combination of multiple methods (e. g. rule-based systems and combinatorial optimization), we contribute a novel formulation that solves layout adaptation using combinatorial optimization only. In contrast to Lindlbauer et al.'s [32] three-stage optimization approach, our approach relies on a single global optimization. Our consideration is that involving multiple stages may encode biases and constraints into the search space, hence running the risk of getting locally optimal results. For instance, in Lindlbauer et al. [32], if a virtual element is set as view-anchored (a stage 1 decision), it is constrained to a low level of detail (a stage 2 decision). Highly relevant virtual elements might therefore not be displayed with all necessary information due to early-stage decisions. In our approach, all decisions are made simultaneously. This arguably enables more flexible traversals of the search space. Our approach is also more scaleable as sub-objectives can easily be added and adjusted in weighting.

2.3 Environment-aware MR

SemanticAdapt is based on the observation that there exists a strong semantic connection between virtual interface elements and physical objects. To enable this approach, our implementation and future renditions rely on 3D reconstruction and semantic segmentation of environments. Commercial tools, like *Canvas* [37] or *Polycam* [25] on the iPhone 12 Pro and the *Microsoft Hololens*, as well as research like KinectFusion [26] and DepthLab [10], have significantly lowered the barrier of entry to accessing environmental geometry data. Likewise, substantial research on object detection and semantic segmentation (e. g. Redmon et al. [42], Duan et al. [11], Valentin et al. [51], Song et al. [46]) has enabled applications to obtain a semantic understanding of the user's physical surroundings.

Substantial work in VR leverage environment geometry data to generate custom virtual environments [48]. Both Shapira et al. [44] and Sra et al.'s [47, 48], for example, use depth information to pair physical objects and virtual counterparts to achieve more immersive VR experiences. Cheng et al. [7] used such mapping for redirected walking in unknown environments. Rather than generating virtual environments, Lindlbauer and Wilson [34] leveraged 3D reconstructions to enable manipulations to the physical environment usually only available to virtual contexts. In AR contexts, the environment geometry is often used for perspective-corrected renderings [14, 27, 33, 41, 43] and anchoring virtual elements [36]. Leveraging geometric information about the environment, Fender and Müller [16] identify the state of the room and adapt the virtual content accordingly.

Several works additionally analyze the affordance of physical objects in the environment. Hettiarachchi and Wigdor [24] presented a system that opportunistically annexes physical objects to provide haptic sensations for virtual objects. Lin et al. [31] employed an object detection model to overlay virtual proxies to physical objects with similar affordances (e. g. a virtual tree stump to a physical chair). Pejsa et al. [41] similarly leveraged labeled physical object affordances to place remote avatars. Our work focuses on leveraging information on the semantic associations between virtual elements and physical objects to support MR layout adaptation across different environments.

3 FORMATIVE STUDY: USER-DESIGNED MR LAYOUTS

We conducted a formative study to characterize user considerations and strategies for designing and adapting MR layouts. Twelve Participants manually generated and adapted the layout of virtual interface elements in different environments. Based on the observations and participants' feedback, we present a set of requirements for automatically adapting MR layouts, which we employ in the optimization scheme.

3.1 Tasks

Participants first *created* the layout of a set of virtual interface elements for one environment, and then *adapted* the layout for another environment. This procedure was repeated twice for two scenarios, i. e. once per scenario. Participants could adjust the virtual interface elements using standard select, rotate, translate, and scale controls. Before the initial generation of the layout, the virtual interface elements were placed in a randomized grid beside participants.

3.2 Scenarios, environments and virtual interface elements

Participants performed the experiment tasks (*create* and *adapt*) for two different scenarios, *productivity* and *leisure*. The scenarios were chosen since they are typical for MR usage, as discussed in previous work [19, 32]. For *productivity*, participants were instructed to imagine themselves as a graduate student surveying papers for a given topic. For *leisure*, participants were instructed to consume online information such as news while chatting with a peer. This scenario represents a separate class of casual daily activities, where efficiency may not be the first priority.

For *productivity*, we tested two environments with participants sitting in front of a desk either in a bedroom (P1) or an open office (P2). For *leisure*, participants were sitting on a sofa in the living room (L1) or on a chair in a coffee shop (L2). The four environments (shown in Figure 2) differed in their visual complexity, social setting and available space. Presentation of the scenarios and the environments was counterbalanced.

For each scenario, participants were asked to work with a different set of virtual interface elements, representing typical applications used in the corresponding activities. They were designed as either 2D panels or 3D visualizations or models, as shown in Figure 3. *Productivity* contained virtual interface elements for websites, documents, images, as well as several 3D visualizations and a 3D model. *Leisure* contained 2D panels for news feeds, shopping, social media, and a calendar, and 3D widgets for weather and time. Participants determined the placement of these elements with the expectation to interact with them, but they did not actually perform the interaction during the experiment.

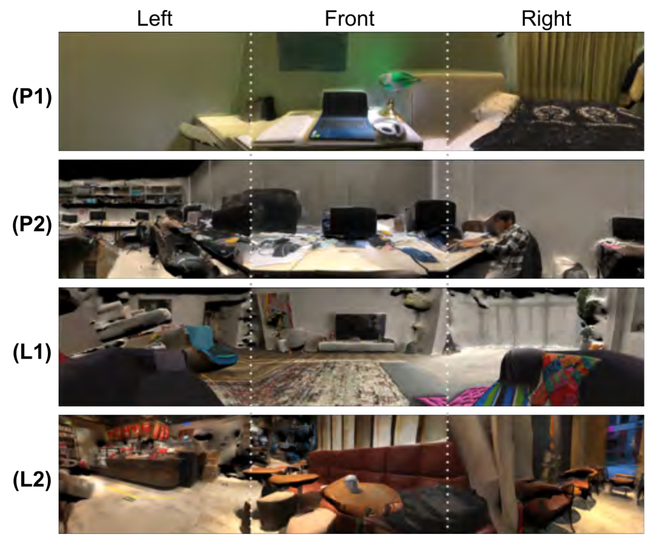

Figure 2: From top to bottom: environments P1, P2, L1, L2 where participants were tasked with either creating or adapting MR layouts in the formative study.

3.3 Participants and apparatus

We invited twelve participants from a local university (7 male, 5 female; age: $M = 22.25$, $SD = 1.33$). All had prior experience with using AR and VR devices with $M = 4.58$, $SD = 0.99$ on a scale from 1 (not at all familiar) to 7 (extremely familiar). The experimental application was developed in Unity 2019 for the Oculus Quest headset. The environments were shown in VR as 3D scans captured using an iPhone 12 (*Canvas* [37]). This decision was made to enable a larger field of view and no lag between a camera and the headset, and to increase internal validity of the experiments since the environments were unchanged between participants. Participants could arrange the layout using Oculus Touch controllers. All sessions were video recorded using SideQuest [45].

3.4 Procedure

After completing the consent form and demographic questionnaire, participants performed a training session to get familiar with the layout controls. The experimenter then introduced the first scenario and environment, and participants started to generate the layout. Participants were instructed to create a world-embedded MR layout that can facilitate the scenario. Once they confirmed the layout met this requirement, they were virtually moved into the second environment, and were tasked with adapting their previous layout to the new space. Participants could view a snapshot of their previous layout by pushing a controller button. This "create-adapt" procedure was performed for both scenarios. During the task, participants

Figure 3: Sets of virtual interface elements that participants were tasked with placing into a layout in the *Productivity* and *Leisure* scenarios in the formative study.

were asked to follow a think-aloud protocol, which was fostered by the experimenter. After completion of all tasks, we conducted a semi-structured interview on (1) motivations and strategies for element placements, (2) perceived environmental factors for placement decisions, and (3) influence of the previous layout on layout adaptation.

3.5 Results

Based on the notes and the participant think-aloud comments, two authors coded the data into 576 segments of individual element placements. We then synthesized a framework describing how participants create and adapt MR layouts for different environments. We specifically focused on identifying potential factors and requirements that should be considered in an MR layout adaptation pipeline.

3.5.1 Overview. Participants typically first performed a *spatial segmentation* of the environment - checking for space that is available for element placements. They then completed two parallel subtasks: (1) searching for the ideal placement for each individual element and (2) optimizing the entire layout. For the first subtask, they considered the following aspects: *semantic associations* between virtual elements and physical objects, the *utility* of each virtual element for their current task, and *compatibility* between virtual elements and the spatial properties of potential placement locations. For the second subtask, they took into account the *layout appearance*, *consistency* between their original and adapted layouts, and ensuring the entire layout to be *visible*.

3.5.2 Spatial segmentation. Participants typically segmented the environment into areas that they consider appropriate for placing virtual interface element. Besides the size of the physical space, participants (N=9) considered the presence of other people and the typical usage of the space (N=3) as factors. Participants tended to avoid occupying space belonging to other people. They additionally matched the typical usage of the spaces with the virtual element. For example, two participants stated that they avoided placing productivity-related elements on top of the bed, because the space usually signifies a function of rest (P7, P11). Additionally, in a public environment such as the coffee shop, participants were more conscious of the amount of space their layout occupied (N=7) and felt inclined to place private elements (e. g. messengers) in secluded locations.

3.5.3 Semantic associations. Participants exhibited three placement strategies that consider the relationship between a virtual interface element and the environment (Figure 4). We refer to the strategies as *anchor*, *avoid*, *replicate*.

- **Anchor**. Physical surfaces (e. g. walls, tables, monitors) were frequently used anchors for the placement of elements (N=11). Additionally, participants "connected" virtual and physical object with direct semantic relation (N=9), such as a virtual time widget to a physical clock (P2) or a virtual paper to a physical notebook (P8). Lastly, participants often anchored the whole layout around a *primary physical object* (N=9), such as physical display (office) (N=10) or TV (living room) (N=7). While there was no direct relation in terms of

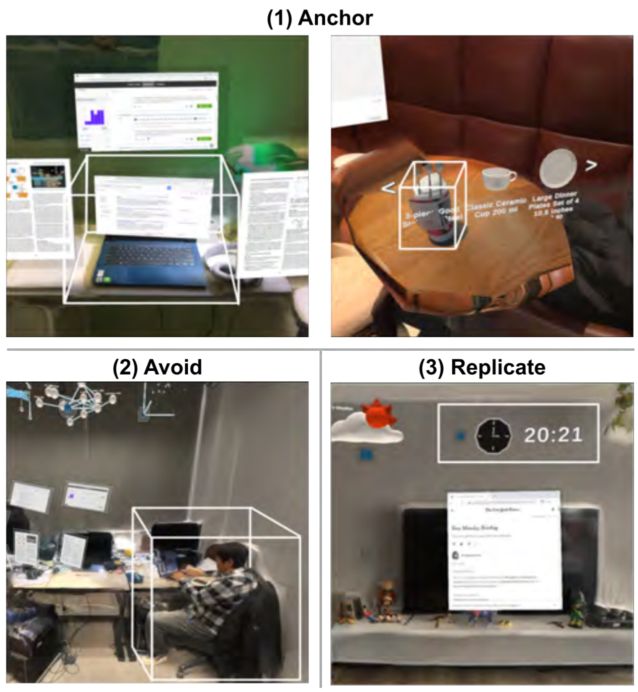

Figure 4: Examples of participant placements influenced by semantic associations. *(1)* presents two examples of *anchoring*. The left image under *(1)* highlights the usage of a laptop computer as a generic anchor. The right image under *(1)* shows the placement of a kitchen utensils shopping application next to a physical coffee cup as a semantic anchor. *(2)* is an example of *avoidance*. The user specifically distances the virtual interface elements from their colleague. *(3)* shows an instance of the *replicate* behaviour. The user places a virtual time widget where they expect a physical clock to be otherwise located.

content, this generic physical anchor object was often core to participants' MR layout.

- **Avoid**. Participants consistently avoided any intersection between virtual elements and physical objects (N=12), as well as placing virtual elements between themselves and other people in the environment (N=9). Participants also avoided cluttered physical backgrounds due to visibility concerns (N=7).
- **Replicate**. Virtual elements were used to replicate or replace absent physical objects (N=8). This included a chat window that was placed on a sofa to serve as stand-in for a peer sitting next to them (P12), or the weather widget placed high up in the environment, replicating the sun (P3).

3.5.4 Utility considerations. A key factor users considered in determining the placement of virtual interface elements was their "relevance" (N=8) or "expected frequency of use" (N=7) with respect to the current task. We use the term *utility* to represent this consideration. Participants generally agreed that higher utility elements should be placed in more accessible positions: within reach (e. g. *P10*: "if a virtual element is more relevant [to my current task], I would prefer it closer to me") and within sight (e. g. *P1*: "elements that are important should just be placed in front of me").

3.5.5 Interaction requirement considerations. Users listed two task-independent characteristics of virtual elements that must be considered. These characteristics reflect how each element is typically interacted with. Firstly, they specified that visibility was more important for some elements than others (N=5). For example, text-heavy elements may need to be placed closer to participants out to concerns of *text legibility*. Secondly, they considered whether an element requires frequent touch interaction (N=6). In particular, participants preferred the placement of 3D models and visualizations in open spaces to enable convenient navigation around them.

3.5.6 Layout considerations. Participants generally preferred structured layouts, and arranged virtual interface elements into rows and columns. A special case we observed was that one participant (P9) dedicated a surface as container, and arbitrarily placed elements in it without inherent meaning. This indicates that while in general, suggesting structured layouts is valuable, users should retain the freedom to arrange virtual elements freely.

We furthermore observed participants forming spatial groupings of virtual elements based on four factors: (1) content similarity (e. g. paper and a related figure), (2) type (3D elements vs. planar elements), (3) relevance to the current task, and (4) interaction requirements. We found one prominently repeated grouping based on *legacy bias* - participants had a tendency of grouping the time and weather widgets together because the two pieces of information are often displayed together on smartphones. Our approach takes those factors as user preference into account.

3.5.7 Historic data. Participants were strongly influenced by prior layouts they had created for different environments. Several participants explicitly stated that consistency with their previous design was a goal they optimized for, and especially prioritized virtual interface elements with high utility. They reasoned that some *personal habits* should be invariant to the environment. They also attributed their expectation for consistency to their prior experiences with screen-based devices, where the UI layout rarely changes. To achieve a level of consistency between their layouts in different environments, participants frequently tried to exploit environmental similarities. For instance, participants would find physical anchors that shared similar features to serve as primary anchors.

3.6 Adaptive MR requirements

Based on the results, we define the following requirements for MR layout adaptation.

- ***R1: Spatial reflection of semantic associations.*** The presence of certain physical objects should influence the placement of virtual interface elements. Elements and objects which share semantic associations should be placed in close proximity. Certain physical objects such as other people in the space should be avoided.
- ***R2: Utility considerations.*** Elements of higher utility for the current task should be assigned to more accessible positions.

- ***R3: Interaction requirements.*** The characteristics and usage requirements of virtual interface elements should be considered for the layout: dimensionality, legibility, and touch interaction requirements.
- ***R4: Layout appearance.*** A structured arrangement of virtual elements in rows or columns was appreciated, similar to grid layouts in 2D interfaces [8].
- ***R5: Spatio-temporal consistency.*** Consistency between the placements of virtual elements in *input environment* and *target environment* should be maintained to enable spatial memory.
- ***R6: Layout visibility.*** Occlusions between elements and objects should be avoided.

4 SEMANTICADAPT

The computational problem of MR layout adaptation can be defined as follows: given a set of virtual elements that were laid out by a user in one environment (*input environment*) in a given usage scenario, determine an placement for each element to support the user in the same scenario in a different environment (*target environment*). We propose an approach that adapts the placement of virtual elements in a *target environment* by solving a linear program. In the following, we detail the implementation with inputs, optimization scheme and constraints, illustrated in Figure 5.

4.1 Inputs

Our model has two sources of inputs: *element parameters*, which characterize the set of virtual elements, and *environment parameters*, which describe the physical environment. All inputs are summarized in Table 1.

4.1.1 Element parameters. Our model computes a set of properties for each element $e \in E$. This includes each element's semantic associations, utility for a given task, and interaction requirements (e. g. touch enabled). While some properties are currently defined manually by content creators, we believe that it is possible to determine these parameters automatically in a data-driven manner. Finding a valid source for this data, however, will be challenging and require longitudinal deployment or investigating if transferring data from other sources (e. g. smartphones) yields desired results.

Semantic parameters. To compute a layout that is semantically compatible with the physical environment, we define a set of related semantic associations a_e^+ (*anchor semantic associations*) and a set of semantic associations to avoid a_e^- (*avoidance semantic associations*) for each element. For example, we define the following related semantic associations for a weather application: {sun, temperature, outdoors, rain}. We use the semantic association lists a_e^+, a_e^- to determine element placements in relation to environment objects.

Utility. Since the results of formative study indicate that an element's placement is highly dependent on its task-specific usefulness, we define a utility value u_e for each element, similar to prior work [32, 40].

Table 1: Description of input parameters.

Element parameters	
Parameter	Description
$E = (e_1, ..., e_n)$	Set of virtual interface elements to place in the layout
$N_e \in \mathbb{Z}^+$	Number of virtual elements in E
$d_e \in \{0, 1\}$	Whether element e is $2D$ or $3D$
$\mathbf{p}_e \in \mathbb{R}^3$	Position of element e in the input environment
$\mathbf{s}_e \in \mathbb{R}^3$	Size of element e
a_e^+	Anchoring semantic associations of element e
a_e^-	Avoidance semantic associations of element e
$u_e \in [0, 1]$	Utility of element e for a given task
$v_{e,vis} \in [0, 1]$	Visibility requirements of element e
$v_{e,touch} \in [0, 1]$	Touch interaction requirements of element e
$v_{e,bg} \in [0, 1]$	Background complexity tolerance of element e
Environment parameters	
Parameter	Description
$O = (o_1, ..., o_n)$	Set of considered physical objects in the environment
$N_o \in \mathbb{Z}^+$	Number of physical objects in O
t_o	Classification of environment object o
$\mathbf{p}_o \in \mathbb{R}^3$	Position of physical object o
$\mathbf{s}_o \in \mathbb{R}^3$	Size of physical object o
$u_o \in [0, 1]$	Whether object o is a good general anchor for virtual elements
$C = (c_1, ..., c_n)$	Set of potential containers to place virtual interface elements
$N_c \in \mathbb{Z}^+$	Number of containers in C
$d_c \in \{0, 1\}$	Whether container c is $2D$ or $3D$
$\mathbf{p}_c \in \mathbb{R}^3$	Position of container c in the input environment
$\mathbf{s}_c \in \mathbb{R}^3$	Size of container c

Interaction requirement parameters. We define several variables representing spatially-dependent interaction requirements. We consider if an element needs to be visible for a given task, represented as $v_{e,vis}$, and whether touch interactions are required, denoted as $v_{e,touch}$. Our approach also considers whether an element's legibility is impacted negatively by visual background complexity (cf. Laramee & Ware [30]), denoted as $v_{e,bg}$. Additionally, the model contains the dimensionality d_e (i. e. $2D$ or $3D$), size $\mathbf{s}_e$, and position in the input environment $\mathbf{p}_e$ of each virtual element. We use dimensionality d_e and size $\mathbf{s}_e$ to constrain an element's potential placements, specifically preventing overlaps between elements and the physical environment in generated layouts. We use the position variable $\mathbf{p}_e$ to optimize for spatio-temporal placement consistency between the layouts of the *input environment* and the *target environment*.

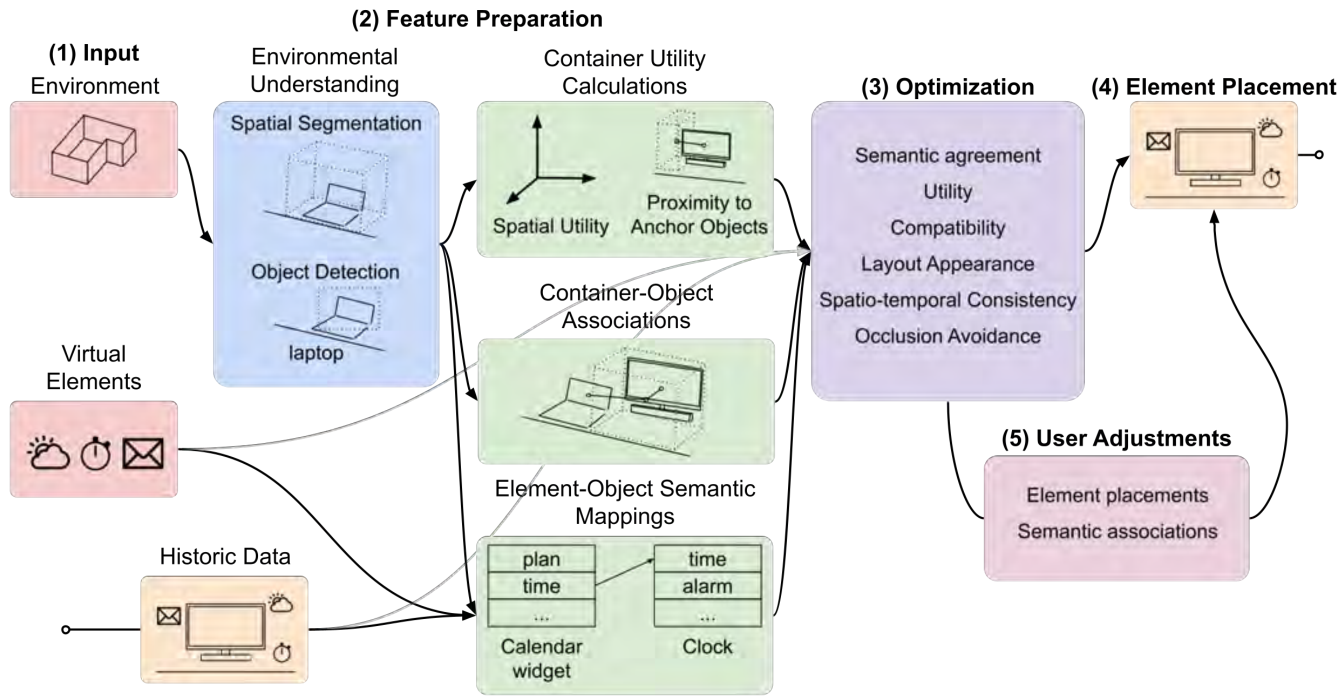

Figure 5: Overview of the proposed approach. We propose a five-stage pipeline for adapting MR layouts between environments: *(1) input* - leverage knowledge of the environment and the virtual interface elements *(2) feature preparation* - from the input environment and element set, extract relevant element and environment parameters. *(3) optimization* - formulate the problem of placing virtual elements in the physical environment as a linear program; *(4) element placement* to form a layout; *(5) user adjustment* - users can optionally adjust the element placements or the semantic associations inferred by the model.

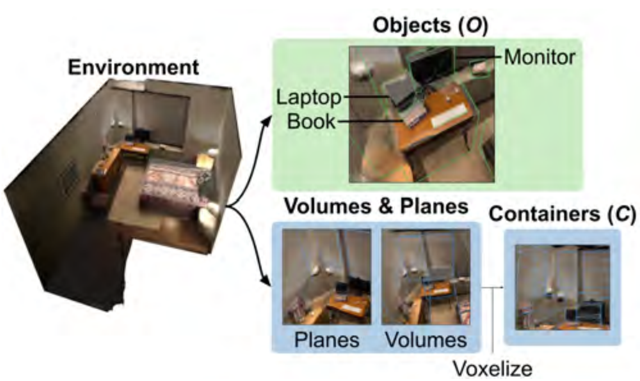

Figure 6: SemanticAdapt extracts two types of information from the environment to inform its placement of virtual interface elements: physical objects O and potential element placement containers C.

4.1.2 *Environment Parameters.* The model uses information of the *input environment* and the *target environment* to determine the placement of virtual elements. Each environment is described as a set of physical objects O and a set of potential element placement containers C, as illustrated in Figure 6.

For each object $o \in O$ in an environment, the model contains information about its type t_o (e. g. *laptop*, *plant*, etc.). This information was labeled manually, taking one author around 3-5 minutes per environment. This process could presumably be sped up with more advanced interactions (e.g. with SemanticPaint's brushing functionality [51]) and complemented in the future with automated approaches (e.g. Mask R-CNN [23]). Since the results of the formative study indicate that there are certain objects that participants selected repeatedly as anchors, we additionally define an anchoring utility u_o for each object. This defines the extent to which an object acts as a generic anchor for virtual elements, i. e. anchor without semantic connection. Examples include a monitor on a desk or a TV on the wall, which were participants' focus points for a given task. u_o is currently manually defined based on the study results, but could be automatically extracted from historic data, i. e. searching for objects that "attract" virtual elements without apparent semantic connection.

Containers C are defined semi-automatically. We first manually specify potential volumes and planar surfaces in the environment where elements can be placed. The model then computes C by subdividing the volumes and planar surfaces into voxels based on the dimensions of the smallest virtual element, and then recursively merging neighboring containers to identify additional placement possibilities. For each container $c \in C$, the model computes its dimensionality d_c (i. e. $2D$ or $3D$), size $\mathbf{s}_c$, and position $\mathbf{p}_c$.

4.2 Optimization scheme

The core decision of the approach is how to assign virtual elements e to potential containers c:

$$x_{e,c} = \begin{cases} 0 & \text{if } e \text{ assigned to } c \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

After this decision, elements are centered in their assigned containers and oriented to either face the user or conform to the container surface. We propose an objective with the following sub-objectives: semantic agreement (S), utility (U), element-container compatibility (Q), occlusion avoidance (V), layout appearance (A), and spatio-temporal consistency (T).

S measures whether semantic associations are reflected spatially in the layout (**R1**), U measures the utility of the overall layout (**R2**), Q measures whether element requirements are compatible with container characteristics (**R3**). A measures whether elements are arranged in a structured manner (**R4**), T measures consistency between the layout of *input environment* and the generated layout for *target environment* (**R5**), and V is a measure of the visibility of overall layout (**R6**). Our optimization seeks to maximize the overall objective function of the sub-objectives as a weighted sum:

$$\underset{e,c}{\arg\max}\, w_s \cdot S + w_q \cdot Q + w_u \cdot U + w_a \cdot A + w_t \cdot T + w_v \cdot V \quad (2)$$

We empirically determined the weights to be $w_s = 0.25$, $w_q = 0.0625$, $w_u = 0.125$, $w_a = 0.0625$, $w_t = 0.25$, $w_v = 0.25$.

4.2.1 Semantic agreement (S). We aim to generate layouts that spatially reflect the semantic associations between interface elements and the physical environment. To achieve this, our approach places virtual elements in close proximity with objects that share similar associations (a_e^+), while distancing elements from objects with associations that should be avoided (a_e^-). We first compute the semantic anchoring $a_{e,o}^+$ and avoidance $a_{e,o}^-$ between a virtual element e and an object o based on the object's type t_o. To achieve this, we leverage the WordNet lexical database [35]. Our current approach equates the semantic similarity of two terms to the path similarity between their synsets according to WordNet's is-a (hypernym/hyponym) taxonomy. For each element, we compute $a_{e,o}^+ \in [0,1]$ as the maximum path similarity between any term in its anchoring semantic associations a_e^+ and object label t_o. $a_{e,o}^- \in [0,1]$ is computed as the maximum path similarity between any term in its avoidance semantic associations a_e^- and object label t_o.

To place elements e close to objects with strong semantic associations (i. e. high $a_{e,o}^+$ value) and at a distance from objects it should avoid (i. e. high $a_{e,o}^-$ value), we define the sub-objective as

$$S = \frac{1}{N_e}\left[w_{s+}\sum_e\sum_c x_{e,c}\max_o\left(\frac{a_{e,o}^+}{\delta_{c,o}^2}\right) - w_{s-}\sum_e\sum_c x_{e,c}\max_o\left(\frac{a_{e,o}^-}{\delta_{c,o}^2}\right)\right] \quad (3)$$

Where $\delta_{c,o}$ denotes the distance between container c and object o, and N_e denotes the overall number of virtual elements. The main terms are weighted as $w_{s+} = 0.375$ and $w_{s-} = 0.625$. The weights were chosen to reflect a behaviour we observed in our formative study, where in the office scenario, users prioritized ensuring no elements were placed around the neighboring researcher over anchoring elements to objects in the region with shared semantic associations.

4.2.2 Utility (U). In the formative study, we made two key observations about how an element's utility u_e should influence its placement . First, users regarded some spaces in the environment as more useful than others for placing virtual elements. Second, a space's usefulness is partially dependent on its position relative to the user. In the formative study, participants found multiple characteristics "useful", including *within reach*, *within sight*, and *directly accessible*. To encode this consideration into our model, we define a spatial utility $u_{c_{spa}}$ for each container c. Containers represent different positions available for placement in an environment, and hence have an associated utility. We define $u_{c_{spa}}$ as

$$u_{c_{spa}} = \frac{1}{3}\left[e^{-(\delta_c-0.5)^2} + (1+e^{10\cdot(|\delta_{c_y}|-0.5)})^{-1} + u_{c_{dir}}\right] \quad (4)$$

δ_{c_y} denotes the container's vertical distance from the user, and $u_{c_{dir}}$ is defined as

$$u_{c_{dir}} = \begin{cases} 1 & \text{if } c \text{ is in front of the user} \\ 0.5 & \text{if } c \text{ is left or right of the user} \\ 0 & \text{if } c \text{ is behind the user} \end{cases} \quad (5)$$

The function defining $u_{c_{spa}}$ assigns low utility values to containers that are far away from the user or too close for elements to be legible, and high utility values to closer containers. Furthermore, high utility values are assigned to containers vertically situated around eye-level, and decreasing utility values with increasing vertical distance. Lastly, high utility values are assigned to containers in front of the user, medium utility values to containers to the left or right, and low utility values for spaces behind the user.

In addition to a container's position, its utility is also determined by whether it is close to an object with high anchoring utility u_o. We therefore define a combined utility value u_c for each containers c as

$$u_c = u_{c_{spa}} + \max_o\left(\frac{u_o}{\delta_{c,o}}\right) \quad (6)$$

The resulting utility term U is then define as combination of the utility values of the virtual element and the container as

$$U = \frac{1}{N_e}\sum_e\sum_c x_{e,c}\left(w_{u_{max}} \cdot u_e \cdot u_c + w_{u_{eq}} \cdot \frac{e^{-|u_e-u_c|} - e^{-1.0}}{1-e^{-1.0}}\right) \quad (7)$$

with the weights $w_{u_{max}} = 0.6$ and $w_{u_{eq}} = 0.4$. We designed this term as a trade off to optimize for element placements that maximize the overall utility and assigning elements to containers with similar utility values (i. e. $u_e \approxeq u_c$). We introduce this trade-off because we noticed that if the model only optimized for a maximum utility, it would greedily assign all elements to containers with the highest possible utility values u_c, resulting in crowded, undesirable layouts.

4.2.3 Compatibility (Q). Compatibility specifies whether the placement of a virtual element fulfills the requirement of users' current task in terms of visibility $v_{e,vis}$, touch requirements $v_{e,touch}$ and tolerance for background complexity $v_{e,bg}$. For visibility and touch, we determine if a potential container is compatible exploiting our knowledge of the geometry of the environment. We compute the visibility $v_{c,vis}$ of a container c as

$$v_{c,vis} = \begin{cases} 0 & \text{if } c \text{ occluded} \\ \text{smoothStep}\left(0.5, 1.0, \arccos\left(\omega_c\right)\right) & \text{otherwise} \end{cases} \tag{8}$$

ω_c is computed as the angular difference between the user's current viewpoint and the element's position. We chose this function so $v_{c,vis} = 1$ when $\omega_e = 0°$, and $v_{c,vis} = 0$ when $\omega_e = 60°$, and smoothly interpolated in between. This function can be adapted based on the field of view of the employed hardware. We check for occlusion by performing a raycast from the user's point of view to the container and intersecting the ray with the environment geometry.

A containers touch capabilities $v_{c,touch}$ are calculated as

$$v_{c,touch} = (1 + e^{10\cdot(\delta_c - 1)})^{-1} \tag{9}$$

δ_c denotes the distance between the container c and the user. We chose this function so $v_{c,touch} = 1$ when the container is close and decreases to 0 past a certain distance threshold, indicating it is beyond the user's reach.

To determine the background complexity $v_{c,bg}$ of a container, our system first captures an image from the user's point of view. $v_{c,bg}$ is then compute as percentage of the image is occupied by edges (identified with Canny edge detection), across all containers in the environment.

The overall element-container compatibility is then computed as

$$Q = \frac{1}{N_e} \sum_e \sum_c x_{e,c}[v_{e,vis} \cdot v_{c,vis} + v_{e,touch} \cdot v_{c,touch} + v_{c,bg} \cdot v_{e,bg}] \tag{10}$$

This function rewards placements of elements with high visibility requirements in highly visible containers, placements of elements with high touch interaction requirements in containers with high touch support, and punish placements of elements with low background complexity tolerance in containers with high background complexity.

4.2.4 Appearance (A). We reward the structured placement of virtual elements in rows (side-by-side) and columns (top-and-bottom). We accomplish this by defining an additional variable that tracks whether pairs of neighboring containers (denoted as $c_{neighbor}$) that are either side-by-side or top-and-bottom are simultaneously occupied, denoted as

$$x^{struct}_{c,c_{neighbor}} = \begin{cases} 0 & \text{if } \sum_e x_{e,c} \cdot \sum_e x_{e,c_{neighbor}} = 1 \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

We then define this sub-objective as

$$A = \frac{1}{N_e} \sum_{c,c_{neighbor}} x^{struct}_{c,c_{neighbor}} \tag{12}$$

4.2.5 Spatio-temporal consistency (T). We reward consistency between the layout in the *input environment* and the layout generated for a *target environment*. Starting with an element's user-centric position in an input environment $\mathbf{p}_e$, we first compute a scaled position $\hat{\mathbf{p}}_e$ to ensure that every element is contained in the target environment. We then compute a distance between each container c and $\hat{\mathbf{p}}_e$, denoted as $\delta_{e,c}$. This enables the formulation of the sub-objective as

$$T = -\frac{1}{N_e} \sum_e \sum_c x_{e,c} \delta_{e,c} \tag{13}$$

which aims to maintain consistency of each element's position relative to the user's position.

4.2.6 Occlusion avoidance (V). Occlusion avoidance is handled as part of the objective function instead of a hard constraint since in less spacious environments, occlusions may be inevitable, and should arguably be resolved by users. For each container c, we perform a raycast to determine the set of containers c_{occ} it could potentially occlude if occupied by an interface element. We then defined additional variables to track whether containers and their occlusion are simultaneously occupied, calculated as

$$x^{occ}_{c,c_{occ}} = \begin{cases} 0 & \text{if } \sum_e x_{e,c} \cdot \sum_e x_{e,c_{occ}} = 1 \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

The final sub-objective is defined as

$$V = -\frac{1}{N_e} \sum_{c,c_{occ}} x^{occ}_{c,c_{occ}} \tag{15}$$

4.3 Constraints

We introduce a set of constraints to avoid duplicate elements, inter-element and element-environment intersections, and trivial solutions (e. g. multiple elements assigned to one container). We avoid element duplicates by enforcing.

$$\sum_c x_{e,c} \leq 1, \forall e \tag{16}$$

While there might be exceptions depending on a user's preference, our formulation allows representing duplicate elements using an additional variable with the same parameters (i. e. characteristics and utility settings). Participants in the formative study, however, did not express the desire for duplicate elements.

To prevent inter-element overlaps, we restrict the capacity of our containers to at most one element as

$$\sum_e x_{e,c} \leq 1, \forall c \tag{17}$$

As containers are generated by recursively merging neighboring containers, we additionally have to prevent spatially intersecting containers from being simultaneously occupied. Let $c_1, c_2 \in C, c_1 \neq C_2$, this constrained is formulated as

$$c_1 \cap c_2 \neq \emptyset \implies \sum_e x_{e,c_1} + \sum_e x_{e,c_2} \leq 1 \tag{18}$$

To prevent element-environment intersections, which participants in the formative study indicated to be undesirable, we first require elements to fit within their assigned containers in terms of size, formulated as

$$\mathbf{s}_e > \mathbf{s}_c \implies x_{e,c} = 0 \tag{19}$$

The final constraint ensures that the dimensionality of containers and elements is similar, i. e. prevents the placement of $3D$ elements in $2D$ containers as

$$d_e = 1 \land d_c = 0 \implies x_{e,c} = 0 \tag{20}$$

4.4 Incorporating user input

Our approach enables users to adjust the optimization results in two ways. First, users can manually view and update semantic connections between virtual elements and environment objects. Upon generating a new layout, semantic connections are visualized using a semi-transparent line (see Figure 1). Users can either "break" existing connections or define new ones via simple interactions. Second, users can "freeze" a subset of the elements in their layout, when they would prefer these elements to be placed in approximately the same relative positions in the new environment. The selected elements will not be considered in our optimization procedure. The containers they occupy will be marked as unavailable to prevent overlapping elements.

4.5 Implementation

We implement our model in Unity 2019. Gurobi 9.1 [22] through a Python 3.8 interface was used to solve the integer program formulated above. We enable the Unity application to access the optimization functionality via a websocket. We define components in Unity for adjusting the input parameters. We use the NLTK WordNet interface [4] for the path similarity calculations and OpenCV [5] for computing background complexity. We employ scanned environments presented in an Oculus Quest VR headset for evaluating our approach rather than implementing the pipeline in a see-through AR headset (e. g. Microsoft Hololens) or augmenting a VR headset with a front-facing camera (e. g. as in Lindlbauer et al. [32]).

5 EVALUATION

We evaluated the performance of SemanticAdapt by comparing it to a baseline adaptation method. The baseline method, which we dubbed *UserCentric*, automatically scaled layouts to new environments without taking semantic connections or other sub-objectives into account. We chose this baseline as it goes beyond naive manual placement or adjustment. To avoid biasing participant towards our approach, participants were simply instructed to "test two new methods for MR layout adaptations". We asked participants to first manually create an MR layout for a given scenario, similar to the formative study in a randomly selected environment. Participants then subsequently moved to four new environments. For each environment, the MR layouts were automatically adapted using SemanticAdapt or the baseline. Participants were tasked to evaluate and adjust these layouts until they were satisfied. We compared the generated layouts and the final adjusted layouts, and recorded the number of interactions for manual adjustments to measure the quality of automatic adaptation results. Furthermore, we collected participants' comments and subjective feedback on the generated layouts in a think-aloud protocol and with questionnaires, respectively. In summary, results show that the layout produced by SemanticAdapt reduced the number of manual adjustments by 33% compared to the baseline, and participants perceived them to be more suitable to the environment and reported higher willingness to adopt the proposed approach.

5.1 Participants

We recruited 12 new participants from a local university (8 male, 4 female; age: $M = 22.58, SD = 1.24$). All participants reported having used AR/VR devices before, rated $M = 4.50$ ($SD = 1.38$) on a scale from 1 (not at all familiar) to 7 (very familiar).

5.2 Design

We used a mixed factorial design with *adaptation method* as within-subject independent variable (*UserCentric* and *SemanticAdapt*) and *scenario* as between-subject independent variable (*productivity* and *leisure*). The *environments* in which participants were asked to create and adapt MR layouts were set as a control factor. Participants performed the tasks in four environments, which were randomly selected from a set of seven. The environments were selected to represent typical usage scenarios. Each participant evaluated 2 *adaptation method* × 2 *environments* = 4 layouts in one of the *scenarios*. As dependent variable, we measured the number of interactions (i. e. manual adjustments of the layout) with the virtual interface elements. Participants could move and rotate any interface element as desired with standard interactions using VR controllers. We additionally counted the number of elements that were at different locations or orientations when comparing the automatically adjusted layout and the layout after manual refinement for each environment. This metric implicitly defines the number of misplaced virtual elements. As we applied a think-aloud protocol, we decided against measuring time as performance indicator. To collect qualitative metrics, we asked participants to rate their agreement to four statements from the SUS questionnaire [6] and three additional statements tailored to the experiment ("I easily understood the arrangement of the layout"; "I thought the layout was suitable in the physical environment"; "I thought the layout was suitable in the usage scenario"). SUS statements were rated on a five-point Likert-like scale from 1 strongly disagree to 5 strongly agree, while additional statements were rated on a similar seven-point scale.

5.2.1 Adaptation methods. We compared *SemanticAdapt* against a baseline adaptation method (*UserCentric*). *UserCentric* placed virtual interface elements in the same relative position to the user as their previous placement. The overall layout was then scaled up or down depending on the available space in the new environment relative to the previous environment. The scaling was performed primarily to ensure that elements would still be accessible within the new environment. We chose this baseline as it adapts the MR layout in a user-centric manner, but without taking semantic connections between the virtual elements and physical objects into account.

As the input to SemanticAdapt, we defined the utility values of the virtual elements based on common understanding of each *scenario*. This is similar to when MR content creators would define the usefulness of applications for individual tasks [32]. We allowed participants to adjust these values before the experiment if they had different expectations. All other parameters were kept constant throughout the experiment.

5.2.2 Scenario & environments. We used the same usage scenarios defined in the formative study, *leisure* and *productivity*. We updated the virtual interface elements to increase the variety of applications. For the *productivity* scenario, participants were given the following virtual elements: PDF reader, search engine, data viewer, calendar, time and weather widgets, messenger, music streaming application, word processor, and to-do list application. For the *leisure* scenario,

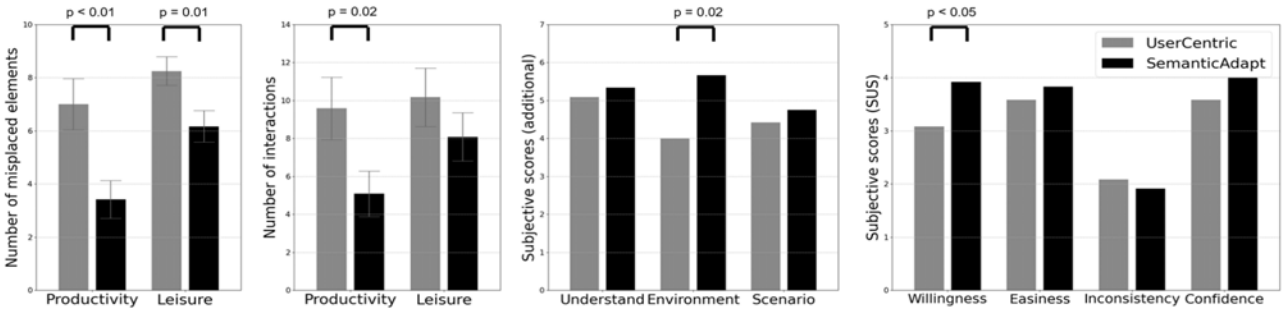

Figure 7: Summary of evaluation results. (a) Number of interactions for manual adjustments. Error bars represent the standard errors. (b) Number of misplaced virtual interface elements. Error bars represent the standard errors. (c) Subjective scores for additional statements tailed to the experiment. (d) Subjective scores for statements from SUS questionnaire.

participants were given a messaging application, time and weather widgets, news reader, photo sharing application, video streaming application, calendar, health tracker, and icon menu. We employed seven different environments in our evaluation. In addition to the four of the formative study, we included a different bedroom, in a small 2 – 4 seat meeting room, in a large conference room. We did not restrict any environment to a particular scenario.

5.3 Apparatus

We used the same apparatus as in the formative study. We simulated the physical environment using a 3D scanned room and conduct the experiment entirely in VR. The experimental platform was implemented in Unity 2019 and deployed on an Oculus Quest. Participants performed the experiment in a seated position. The experiment was conducted in a quiet room, monitored by the experimenter.

5.4 Procedure

After completing a demographics questionnaire, participants were placed in a tutorial scene to learn the platform controls. Participants then manually created an MR layout in one environment, where all virtual interface elements were placed randomly aside them. Afterwards, they were provided with an adapted MR layout in a new environment, as generated by one of the adaption methods. Participants then refined the layout until they were satisfied, i. e. they felt it be useful for the scenario. The adaptation task was repeated for a total of four environments with counterbalanced choice of *adaptation method*. The study concluded with a questionnaire and a semi-structured interview, typically lasting 30 minutes.

5.4.1 Data collection. All sessions were video recorded using SideQuest. Data was collected via logging events in Unity. We counted the number of interactions to adjust the virtual elements manually by reviewing the video footage. This was done since we noticed the existence of a small number unintentional adjustments (e. g. moving a element by accident), which were ignored.

5.5 Results

We performed a series of mixed ANOVAs on number of interactions and number of misplaced virtual interface elements, with *scenario* being the between-subject variable, *adaptation methods* and *environment* being the within-subject variables. Subjective scores were analysed using a series of Wilcoxon signed-rank tests. We performed our data analyses using SPSS Statistics 27.

5.5.1 Quantitative results. Results from the mixed ANOVAs showed a main effect for *adaptation methods* on the number of interactions for adjustments ($F_{1,10} = 7.28, p < 0.05$), and the number of misplaced virtual interface elements ($F_{1,10} = 11.83, p < 0.01$), summarized in Figure 7. On average, *SemanticAdapt* required the participants to adjust 4.79 elements (*SD* = 0.54) with 6.58 interactions (*SD* = 1.07), while *UserCentric* required to adjust 7.63 elements (*SD* = 0.73) with 9.88 interactions (*SD* = 1.44). The results indicated that the layouts generated by *SemanticAdapt* were closer to the optimal layouts in participants' mind than those generated by *UserCentric*. Thus, participants needed to take fewer adjustments to meet their satisfaction. In two occasions, two participants decided to not adjust any element from the layout generated by *SemanticAdapt*. No significant interaction effects between the between-subject variable *scenario* and within-subject variables was found. *Environment* did not not have a statistically significant influence on the dependent variables.

5.5.2 Qualitative Results. A series of Wilcoxon signed-rank tests on the questionnaire data revealed that the layouts generated by *SemanticAdapt* (*M* = 5.67, *SD* = 0.78) were perceived to be more suitable to the environment compared to those generated by *UserCentric* (*M* = 4.00, *SD* = 1.60), Z = -2.39, p = 0.02; participants were more willing to frequently use *SemanticAdapt* (*M* = 3.92, *SD* = 0.79) than *UserCentric* (*M* = 3.08, *SD* = 1.08), Z = -2.00, p < 0.05. No other questions yielded significantly different results between the two methods. From the comments, it seems participants preferred *SemanticAdapt*. Several participants were particularly appreciative of the consideration of semantic associations between virtual elements and the physical environment. P5 remarked that, "I believe AR should really enhance reality. Since I should be used to the placement of different physical things in my environment, it makes sense to place applications next to what they are related to."

Some comments on the semantic connections revealed a tension between adjustments and spatial memory. As noted by one participant, "for elements that I expect to use often, I would prefer

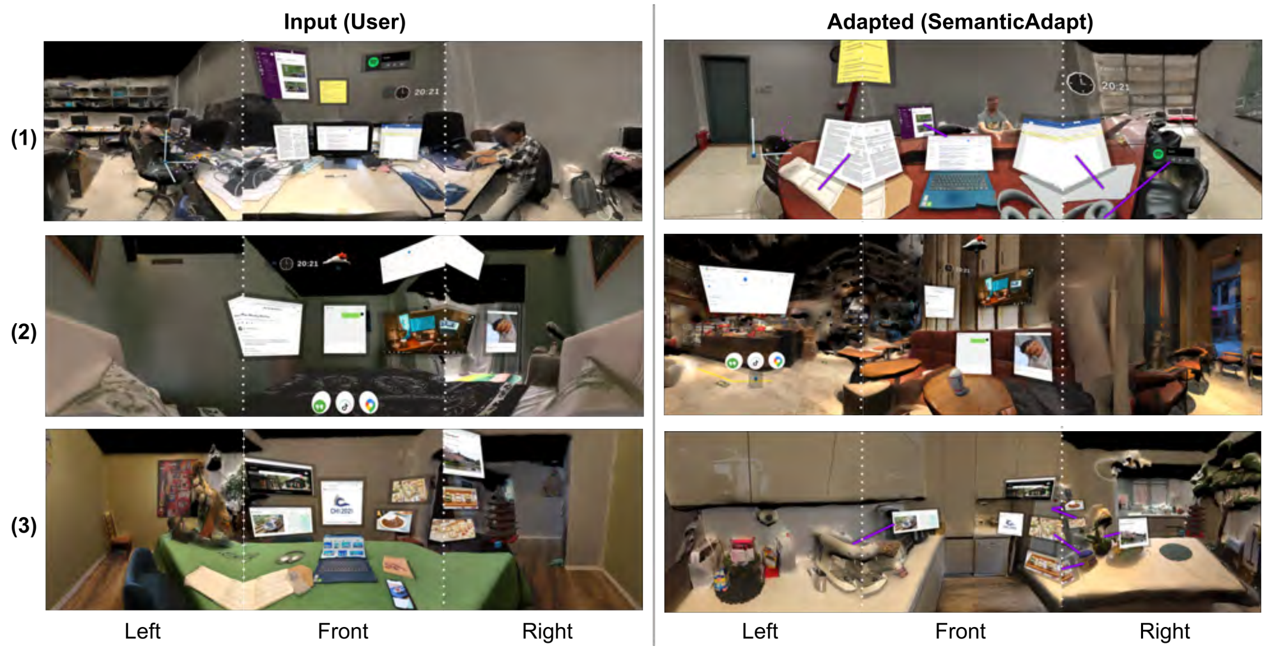

Figure 8: Manually created and automatically adapted interface layouts for the application scenarios detailed in Section 6: (1) conducting research, (2) browsing social media, (3) travel planning. The purple lines in the diagram indicate semantic connections.

them fixed in the same place regardless of what is in the environment" (P1). P3 attributed this preference to the fact that they placed elements in a way that "reflects [their] usage habits" rather than considering the environment. The consideration of semantic connections therefore might be more effective when applied to elements selectively. Specifically, elements with high utility for a task should be available in a highly predictable manner. These comments make us believe that the choice of connection should ultimately be strongly influenced by users' preference. Participants further commented on the specific semantic connections. P4, for example, mentioned that "although semantically placing the time widget next to the clock makes sense, considering how it will be used, it might not be appropriate. The clock is placed where it is for it to be visible to everyone in the office. As the time widget is for my personal use, it should be placed in a more private, accessible location." This interesting observation hints at the fact that users strongly consider privacy and visibility for others, even though virtual elements were only visible to them. Our approach can accommodate for such preferences by considering historic data more strongly, and introducing additional weights for public and private areas in an environment.

6 APPLICATION SCENARIOS

To illustrate the applicability of SemanticAdapt, we demonstrate several example scenarios, shown in Figure 8. Those involve different sets of virtual interface elements and take place in different physical environments. The initial layouts (Fig. 8, *left*) were generated manually, and adapted for the new environments (Fig. 8, *right*) by our proposed approach.

6.1 Portable MR office

We use SemanticAdapt to create a portable MR office. The applications are initially laid out to fit a typical desk worker scenario (Fig. 8, (1) *left*), and include applications like a PDF reader, search engine, and word processor. All virtual interface elements surround the monitor, which serves as primary anchor. The layout is optimized for a individual seated workplace. The portable MR office is then moved to a meeting room (Fig. 8, (1) *right*), and the layout automatically adapted to the new environment. In the new layout, three virtual interface elements were particularly influenced by the semantic associations in the environment. The PDF document (semantic terms: *book*, *reading*, *publication*) is placed next to the notebook. Similarly, the word processor (*report*, *paper*, *document*) is placed close to pile of loose sheets of paper. The music streaming application (*music*, *sound*, *speaker*) was initially placed in the top right corner of the layout. In the new layout, our algorithm shifted the element to table-height next to a pair of headphones.

6.2 Browsing social media

We created a MR layout for casually browsing social media contents, including three news and social media related windows and a central chat applications. Furthermore, the layout contains three icons to access additional applications, a calendar and a weather application, all placed in the user's periphery. The initial layout was created manually in a bedroom (Fig. 8, (2) *left*), and then moved and adapted automatically for a new environment, i. e. a coffee shop (Fig. 8, (2) *right*). In the new environment, the three main windows and the chat application remained centrally located due to their high utility value. The peripheral applications were shifted automatically to the side due to spatial restrictions, i. e. the space directly in front of the user was occupied by the table and other applications. In this scenario, the environment does not offer many physical anchors beside the primary location (table). Therefore, our approach becomes similar to geometry-aware adaptation approaches, focusing on spatial constraints and task relevance of virtual interface elements.

6.3 Travel planning

We created a MR layout suitable for travel planning, manually created for the dining room (Fig. 8, (3) *left*). The layout contains several relevant applications such as a browser with hotel booking website (semantic terms: *sleep*, *bed*, *booking*), a travel website on botanical gardens (*leaf*, *plant*, *nature*), a travel blog on cafes (*coffee*, *espresso*, *seat*) and an image browser (*food*, *snack*, *refrigerator*). Note that the semantic terms were defined on a content rather than an application level, which could be integrated automatically by parsing websites and images. The layout is then moved and adapted for the kitchen (Fig. 8, (3) *right*). As a result, there was significant shared semantic associations between the virtual interface elements and the kitchen appliances. The garden website was, for instance, anchored to a plant. The blog post about a coffee shop was anchored to a coffee machine. The images anchored to either the physical refrigerator or the snack container on the kitchen island table. The placement of applications becomes highly coupled with the semantic context, thus we believe highly memorable and predictable after brief usage periods.

7 DISCUSSION

In this paper, we contributed an optimization-based approach to assist users in adapting MR layouts to new environments. We conducted a formative study in which users were asked to manually create MR layouts for multiple environments and scenarios. We found that users first perform a spatial segmentation of the environment, and then place virtual interface elements based on factors such as semantic mapping, utility and relevance, and several layout considerations. Based on those findings, we present SemanticAdapt, which automatically finds the optimal placement of virtual interface elements, based on semantic associations between virtual interface elements and physical objects, users' current task and environment, as well as prior layouts. We evaluated the performance of SemanticAdapt in comparison to a baseline adaptation method. Results indicate that the proposed approach decreases the number of manual interactions by 33 %, and was preferred in subjective ratings.

7.1 Reference frame: user-centered vs. world-anchored

We focus on the adaptation of world-anchored MR layouts, where virtual interface elements are embedded into the physical environment. In the formative study, however, several users arranged some of elements with a user-centered strategy: elements placements did not consider the environment, but only users themselves. This strategy is in line with prior research on MR placement (e. g. [12, 32]), and would be an interesting addition to our approach. In the simplest version, our approach could be expanded to enable users to "freeze" parts of the layout relatively to themselves in different environments, while arrange the other elements with respect to the environment.

7.2 "Depth" of semantic associations: application vs. content

Currently, we primarily determine the semantic associations between virtual interface elements and physical objects at an application level. The semantic associations are based on the application that the virtual interface element belongs to. For instance, in *productivity* scenario, the PDF reader had the following semantic associations: {reading, concentration, book shelf, magazine, article, news, publication, paper, information} . Those associations, however, could also be defined based on the content that the application displays. In the same example, since the PDF reader shows a paper on mixed reality, the semantic associations could equally be: {augmented reality, virtual reality, computer, technology, future, head mounted display}. We started explored this with the *travel planning* application. We plan on exploring the appropriate depth of semantic associations and methods of combining the use of associations at different levels in the future.

7.3 Model improvement: adaptation and personalization

In the evaluation, we observed several behaviours that indicate a potential need for adaptive models. Since we assumed the utilities of virtual interface elements to be determined by the *scenario*, we kept these values fixed across different environments. We found, however, that some environments also play a role in the utilities of the elements. In an occupied meeting room, for example, one participant avoided using the messenger element, i. e. assigned a lower perceived utility. This did not change even after we explained that the application would not be seen by others. The observation makes us believe that adjusting the model parameters, including utilities of elements, according to the environments has the potential to improve the predictive performance of our approach. The results from the experiments furthermore indicate that the optimal layout differs between users. This is expected, as users mentally assigned different weights to the importance of semantic associations compared to layout consistency and efficiency; and favor different subsets of the semantic associations. This hints at the importance of providing multiple alternative layout options for users to choose from, and personalizing the model iteratively based on their adjustments in different environments. While our approach takes previous layouts into account, we plan to explore even more personalized models.

7.4 Limitations and Future Work

While our evaluation generally indicates that semantic associations merit consideration in adapting MR layouts, we acknowledge there are contexts where its importance is decreased. The contexts include when the conflicting factors dominate the decision, or semantic associations are not adequate or lead to confusion. Identified conflicts include those with layout stability and predictability, interaction requirements, and personal preference. In environments where either multiple anchors share semantic associations or strong semantic associations are almost entirely lacking, this may lead to placements in undesirable locations. Future work exploring contextually weighting semantic associations amongst other factors of MR interface adaptation and evaluating individual contributing factors may reveal interesting insights.

In the current implementation, we require several input variables describing the environment to be manually extracted, specifically the physical objects and potential element containers in the environment. We believe that future versions of our approach will be complemented and replaced by techniques that can extract this information automatically, such as Mask R-CNN [23] for semantic segmentation. We could furthermore leverage recordings (e. g. in OptiSpace [14] or HeatSpace [15]) of users' behaviour automatically determine the utility values of specific spatial areas. The current approach is designed to be modular, allowing more accurate future models to be flexibly integrated. We plan to further explore the directions of supporting annotations of environments for the purposes of MR interface adaptation and developing automated techniques for doing so.

Our formulation of MR interface adaptation as a single global optimization was in part motivated by considerations of extensibility. In contrast to a multi-stage optimization approach, a global optimization formulation arguably enables more flexible traversals of the search space and straightforward ways to add and adjust sub-objectives. We plan to perform a direct comparison between the two optimization approaches to pinpoint the benefits and limitations of their designs. We believe our open sourced implementation will also support the community in exploring this perspective.

Last, our model currently only considers two strategies for semantic mapping: avoid and anchor. In the formative study, multiple participants mentioned replication as a third strategy. We decided against including this in our current approach because without the presence of a real anchor (i. e. the lack of data), locations to replicate the physical object are very challenging to predict. Users can, however, manually position elements to fulfill this need. In the evaluation, participants did not request this feature. We believe that this challenge could be addressed by further exploring users' expectation about the appropriate locations for replication in the future.

8 CONCLUSION

We present SemanticAdapt, an optimization-based approach to adapt MR layouts between different environments. Our approach alleviates the need for repeated manual adjustments in different environments by automatically deciding the placement of virtual interface elements. We see the strength in our approach in specifically supporting users in setting up their interfaces within new environments. The key consideration of SemanticAdapt is to exploit the semantic associations between the virtual interface elements and the physical objects. Maintaining these associations helps achieve semantic consistency of the layout across different environments. SemanticAdapt furthermore considers the utility of elements for users' current task, layout factors, and spatio-temporal consistency to previous environments. All those factors are combined in a single linear program, which is used to adapt the layout of MR layouts in real time. Our evaluation show that SemanticAdapt decreased the number of manual interface adaptations by 33% and was preferred by users. We believe our approach is a step towards tightly and automatically integrating MR layouts into users' physical environment, which we believe is a key factor for creating beneficial and unobtrusive MR interfaces.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 61902208, the grant from the Institute for Guo Qiang, Tsinghua University No. 2019GOG0003, the China Postdoctoral Science Foundation under Grant No. 2021M691788. The authors thank the reviewers for their suggestions on improving this paper, all participants for their time and effort.

REFERENCES

[1] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. 2001. Recent Advances in Augmented Reality. *IEEE Comput. Graph. Appl.* 21, 6 (Nov. 2001), 34–47. https://doi.org/10.1109/38.963459

[2] Ronald Azuma and Chris Furmanski. 2003. Evaluating Label Placement for Augmented Reality View Management. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)*. IEEE Computer Society, Washington, DC, USA, 66–. http://dl.acm.org/citation.cfm?id=946248.946790

[3] Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. ACM, New York, NY, USA, 101–110. https://doi.org/10.1145/502348.502363

[4] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[5] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[6] John Brooke. 1996. Sus: a "quick and dirty'usability. *Usability evaluation in industry* 189 (1996).

[7] L. P. Cheng, E. Ofek, C. Holz, and A. D. Wilson. 2019. VRoamer: Generating On-The-Fly VR Experiences While Walking inside Large, Unknown Real-World Building Environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 359–366. https://doi.org/10.1109/VR.2019.8798074

[8] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. GRIDS: Interactive Layout Design with Integer Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376553

[9] Stephen DiVerdi, Tobias Höllerer, and Richard Schreyer. 2004. Level of Detail Interfaces. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*. IEEE Computer Society, Washington, DC, USA, 300–301. https://doi.org/10.1109/ISMAR.2004.38

[10] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. DepthLab: Real-Time 3D Interaction with Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 829–843. https://doi.org/10.1145/3379337.3415881

[11] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. CenterNet: Keypoint Triplets for Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[12] Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. 1993. Windows on the World: 2D Windows for 3D Augmented Reality. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology* (Atlanta, Georgia, USA) *(UIST '93)*. Association for Computing Machinery, New York, NY, USA, 145–155. https://doi.org/10.1145/168642.168657

[13] Anna Maria Feit. 2018. *Assignment Problems for Optimizing Text Input.* Ph.D. Dissertation. Aalto University. http://urn.fi/URN:ISBN:978-952-60-8016-1

[14] 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*

(Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173843

[15] Andreas Fender, David Lindlbauer, Philipp Herholz, Marc Alexa, and Jörg Müller. 2017. HeatSpace: Automatic Placement of Displays by Empirical Analysis of User Behavior. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 611–621. https://doi.org/10.1145/3126594.3126621

[16] Andreas Fender and Jörg Müller. 2019. SpaceState: Ad-Hoc Definition and Recognition of Hierarchical Room States for Smart Environments. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (Daejeon, Republic of Korea) *(ISS '19)*. Association for Computing Machinery, New York, NY, USA, 303–314. https://doi.org/10.1145/3343055.3359715

[17] Krzysztof Gajos and Daniel S. Weld. 2004. SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (Funchal, Madeira, Portugal) *(IUI '04)*. Association for Computing Machinery, New York, NY, USA, 93–100. https://doi.org/10.1145/964442.964461

[18] 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.

[19] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 197–208. https://doi.org/10.1145/3332165.3347933

[20] Raphael Grasset, Tobias Langlotz, Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. 2012. Image-driven View Management for Augmented Reality Browsers. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (ISMAR '12)*. IEEE Computer Society, Washington, DC, USA, 177–186. https://doi.org/10.1109/ISMAR.2012.6402555

[21] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (June 2017), 1706–1724. https://doi.org/10.1109/TVCG.2016.2543720

[22] LLC Gurobi Optimization. 2021. Gurobi Optimizer. http://www.gurobi.com

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask R-CNN. arXiv:1703.06870 [cs.CV]

[24] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. *Annexing Reality: Enabling Opportunistic Use of Everyday Objects as Tangible Proxies in Augmented Reality*. Association for Computing Machinery, New York, NY, USA, 1957–1967. https://doi.org/10.1145/2858036.2858134

[25] Polycam Inc. 2021. Polycam. https://poly.cam/

[26] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 559–568. https://doi.org/10.1145/2047196.2047270

[27] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 637–644. https://doi.org/10.1145/2642918.2647383

[28] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, T. Hollerer, and S. Sestito. 2000. Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*. 3–11. https://doi.org/10.1109/ISAR.2000.880917

[29] Andreas Karrenbauer and Antti Oulasvirta. 2014. Improvements to Keyboard Optimization with Integer Programming. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 621–626. https://doi.org/10.1145/2642918.2647382

[30] Robert S. Laramee and Colin Ware. 2002. Rivalry and interference with a head-mounted display. *ACM Trans. Comput.-Hum. Interact.* 9, 3 (2002), 238–251. https://doi.org/10.1145/568513.568516

[31] Chuan-en Lin, Ta Ying Cheng, and Xiaojuan Ma. 2020. ARchitect: Building Interactive Virtual Experiences from Physical Affordances by Bringing Human-in-the-Loop. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376614

[32] 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* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 147–160. https://doi.org/10.1145/3332165.3347945

[33] David Lindlbauer, Jörg Mueller, and Marc Alexa. 2017. Changing the Appearance of Real-World Objects By Modifying Their Surroundings. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3954–3965. https://doi.org/10.1145/3025453.3025795

[34] David Lindlbauer and Andrew D. Wilson. 2018. *Remixed Reality: Manipulating Space and Time in Augmented Reality*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173703

[35] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41. https://doi.org/10.1145/219717.219748

[36] 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* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1233–1244. https://doi.org/10.1145/2858036.2858250

[37] Occipital. 2021. Canvas. https://canvas.io/

[38] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with Interactive Layout Suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1221–1224. https://doi.org/10.1145/2702123.2702149

[39] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John, and A. Karrenbauer. 2020. Combinatorial Optimization of Graphical User Interface Designs. *Proc. IEEE* 108, 3 (2020), 434–464. https://doi.org/10.1109/JPROC.2020.2969687

[40] Seonwook Park, Christoph Gebhardt, Roman Rädle, Anna Maria Feit, Hana Vrzakova, Niraj Ramesh Dayama, Hui-Shyong Yeo, Clemens N. Klokmose, Aaron Quigley, Antti Oulasvirta, and Otmar Hilliges. 2018. AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173758

[41] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. 2016. Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (San Francisco, California, USA) *(CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1716–1725. https://doi.org/10.1145/2818048.2819965

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[43] Jan Riemann, Martin Schmitz, Alexander Hendrich, and Max Mühlhäuser. 2018. FlowPut: Environment-Aware Interactivity for Tangible 3D Objects. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 31 (March 2018), 23 pages. https://doi.org/10.1145/3191763

[44] L. Shapira and D. Freedman. 2016. Reality Skins: Creating Immersive and Tactile Virtual Environments. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 115–124. https://doi.org/10.1109/ISMAR.2016.23

[45] SideQuest. 2021. SideQuest. https://sidequestvr.com/

[46] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic Scene Completion From a Single Depth Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[47] M. Sra, S. Garrido-Jurado, and P. Maes. 2018. Oasis: Procedurally Generated Social Virtual Spaces from 3D Scanned Real Spaces. *IEEE Transactions on Visualization and Computer Graphics* 24, 12 (2018), 3174–3187. https://doi.org/10.1109/TVCG.2017.2762691

[48] Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie Maes. 2016. Procedurally Generated Virtual Reality from 3D Reconstructed Physical Space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (Munich, Germany) *(VRST '16)*. Association for Computing Machinery, New York, NY, USA, 191–200. https://doi.org/10.1145/2993369.2993372

[49] T. Tahara, T. Seno, G. Narita, and T. 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)*. 249–255. https://doi.org/10.1109/ISMAR-Adjunct51615.2020.00072

[50] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (Brisbane, QLD, Australia) *(DIS '16)*. Association for Computing Machinery, New York, NY, USA, 543–555. https://doi.org/10.1145/2901790.2901817

[51] Julien Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Nießner, Antonio Criminisi, Shahram Izadi, and Philip Torr. 2015. SemanticPaint: Interactive 3D Labeling and Learning at Your Fingertips. *ACM Trans. Graph.* 34, 5, Article 154 (Nov. 2015), 17 pages. https://doi.org/10.1145/2751556