User-Driven Constraints for Layout Optimisation in Augmented Reality

Aziz Niyazov, Barrett Ens, Kadek Ananta Satriadi, Nicolas Mellado, Loïc Barthe, Tim Dwyer, Marcos Serrano

HAL Id: hal-04042351
https://hal.science/hal-04042351

Submitted on 23 Mar 2023

User-Driven Constraints for Layout Optimisation in Augmented Reality

Aziz Niyazov
University of Toulouse 3 - IRIT
Toulouse, France
aziz.niyazov@irit.fr

Barrett Ens
Monash University
Melbourne, Australia
barrett.ens@monash.edu

Kadek Ananta Satriadi
Monash University
Melbourne, Australia
kadek.satriadi@monash.edu

Nicolas Mellado
CNRS - University of Toulouse 3
Toulouse, France
nicolas.mellado@irit.fr

Loïc Barthe
University of Toulouse 3 - IRIT
Toulouse, France
loic.barthe@irit.fr

Tim Dwyer
Monash University
Melbourne, Australia
tim.dwyer@monash.edu

Marcos Serrano
University of Toulouse 3 - IRIT
Toulouse, France
marcos.serrano@irit.fr

ABSTRACT

Automatic layout optimisation allows users to arrange augmented reality content in the real-world environment without the need for tedious manual interactions. This optimisation is often based on modelling the intended content placement as constraints, defined as cost functions. Then, applying a cost minimization algorithm leads to a desirable placement. However, such an approach is limited by the lack of user control over the optimisation results. In this paper we explore the concept of user-driven constraints for augmented reality layout optimisation. With our approach users can define and set up their own constraints directly within the real-world environment. We first present a design space composed of three dimensions: the constraints, the regions of interest and the constraint parameters. Then we explore which input gestures can be employed to define the user-driven constraints of our design space through a user elicitation study. Using the results of the study, we propose a holistic system design and implementation demonstrating our user-driven constraints, which we evaluate in a final user study where participants had to create several constraints at the same time to arrange a set of virtual contents.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Augmented Reality; Spatial interaction; Hand gestures; Constraints; Optimization

1 INTRODUCTION

Augmented reality technologies using see-through head-mounted displays (HMDs) can augment the environment by displaying pervasive content anywhere and anytime. These technologies have

reached an advanced level of maturity in terms of environment tracking, quality of display, and gesture recognition. Recent research has demonstrated that displaying virtual 2D or 3D data around the user and within the real world provides increased spatial understanding thanks to depth cues, decreases the information clutter thanks to the increased display size, and supports more natural interaction techniques such as gestural input and body-based navigation [7]. However, the bottleneck to fully exploit such potential lies in the user interface limitations: in particular, manually arranging this content in the surrounding real world is a very complex and tedious task, yet critical for an efficient access to the data [23].

Previous works have tried to overcome this issue by automatically optimising the content placement in augmented reality environments, removing the need for any user input. The optimisation can be based on different constraints such as semantic association [5], user perspective [11], geometry of the environment [9] or content persistence over time [12]. However, these approaches lack any form of user control over the resulting placement optimisation, even though adding interaction into optimisation systems has been shown to be beneficial and appreciated by users in other contexts [21]. To sum up, there is a need for intermediate approaches in augmented reality combining user input and automatic layout adaptation.

In this paper we address this question of how to bring interactivity into layout optimisation systems for augmented reality environments, by allowing users to guide layout management behaviour while avoiding the need for manual content arrangement. We move away from the inherited UI interfaces that tend to populate augmented reality platforms (e.g. contextual menus or UI widgets), as they break the interaction flow and distract the user's attention from the surrounding real world [17]. Instead, our goal is to propose a holistic, fluid [6] and intuitive [16] interaction approach allowing any user to easily guide the layout optimisation in the real world. Our approach should also be accessible to users with limited expertise in constraint optimisation. This leads us to

address the challenging problem of providing a set of rich spatial gestures that define, all at once, the constraint, its parameters, and its applied spatial region. Our work focuses on laying out 2D widgets as a first step to tackle this challenge, leaving aside other types of virtual content such as 3D widgets or very large 2D windows.

To answer this question, we first review previous optimisation constraints for augmented reality environments, selecting those that can benefit from user control. Then we propose a design space for interactive augmented reality layout optimisation, based on an object-action approach [41] to favor fluid and intuitive augmented reality interaction [6, 16]. Our design space considers the following factors: the user-driven constraint, the constraint application region, and the constraint parameters. To explore this design space, we conducted a gesture elicitation study with 12 participants who proposed gestures for a set of combinations of constraints, parameters and regions while wearing a HMD. Finally, using the results from the study, we designed a system involving a complementary set of gestures to define the various constraints. We developed our design in a proof-of-concept prototype that demonstrates the application of our design space in a potential real-world scenario. Finally, we conducted a user study with 12 participants to validate our approach and gestures, where participants had to create several constraints at the same time to arrange a set of virtual contents.

In summary, we highlight our contributions as follows: 1) A design space of interactive layout optimisation for augmented reality environments; 2) A user elicitation study exploring which gestures would users perform to conduct such interactive optimisation; 3) The design and implementation of an interactive prototype demonstrating our approach; 4) The validation of our approach through a controlled summative study.

2 RELATED WORK

Our contribution relates to previous work on augmented reality layouts and on how to manipulate its content. We also review the existing automatic layout optimisation approaches.

2.1 Augmented reality layouts

In this research we study the display of information spaces in augmented reality. In particular, we address the problem of how to layout multiple 2D widgets, which may be tedious if there are a large number to be moved over a large distance.

Ens et al. [8] introduced a design space for 2D information spaces in augmented reality environments, around a fundamental layout dimension: the reference frame. The reference frame includes the perspective and the movability of the content. In terms of perspective, the content can be arranged in either egocentric (i.e. body-based coordinates) or exocentric (i.e. world-based coordinates) perspectives. In terms of movability, the content can either move with the user or be fixed in space. Most often, the egocentric perspective is combined with movable content, as in the Personal Cockpit or in the Multfi systems [10, 15]. Such body-centric UIs follow the user as they move, thus the organization of the contained widgets is independent from the external environment. Conversely, exocentric UIs often use contain world-fixed content, as when placing documents for collaborative sensemaking [25], or when arranging small multiples visualisations [22]. This later case, i.e. exocentric

perspective with fixed content, has been shown to be relevant for information visualisation, however raises the question of setting the content position by the user. We thus focus on the positioning of virtual 2D widgets in the user's spatial surroundings, i.e. the physical surfaces around her.

2.2 Spatial manipulation

The spatial manipulation of information spaces in augmented reality environments can be classified according to three dimensions [8]: the proximity to the user, the input mode and the tangibility. Proximity describes the distance between the information space and the user: the content can be on the body surface [24, 48], near the user [19, 50], or far away [22, 37]. The input mode can be direct (e.g. direct touch) or indirect (e.g. cursors or ray-casting). Most of the time, direct input [24, 50] is used for near and on-body content, whereas indirect input [25, 37] is used for far content. The third dimension, tangibility, describes whether the information space is mapped to a surface that can be touched. The content can then be either tangible, as when leveraging surfaces such as walls, or intangible, as when displaying the content in mid-air.

However, performing direct or indirect manipulations of content in an augmented reality environment is known to be quite tedious, particularly when the user has to displace multiple information spaces, for instance when transitioning between different real-world environments. Lu and Xu [23] recently conducted a study to understand the limitations of current manual UI transitions approaches in augmented reality: the results show that one of the most common pain-points is the manual placement of UI content within the real world, which requires a high level of effort.

2.3 Augmented reality layout optimisation

To address the pain-point of manually placing content in augmented reality environments, many efforts have been made towards defining automatic layout optimisation approaches. The goal of these optimization approaches is to automatically place the virtual content in desirable locations of the environment to preserve predefined rules (the constraints). This optimization should lead to a good quality layout, i.e. a layout that ensures a good user agency, which can be defined as the extent to which the user feels in control of the final layout [23, 43]. In this section we review the constraints that have been proposed to optimise content placement in augmented reality environments.

Environment geometry: Gal et al. [14] extracted horizontal and vertical surfaces to dynamically arrange virtual objects. Nuernberger et al. [30] extracted the edges of the environment to snap virtual windows to them.

Spatial consistency: maintaining the spatial layout consistency can improve the content memorization across multiple environments [23]. For instance, Ens et al. [9] explored the transition from egocentric to exocentric reference frames while preserving spatial consistency.

Cognitive load: depending on the expected cognitive load of the ongoing task, Lindlbauer et al. [20] increased or decreased the number of UI elements and their level of details.

User perspective: Fender et al. [11, 12] optimised content placement according to the users' Field of View and orientation. Lages

and Bowman [18] explored how to adapt the content to the ongoing location and task while walking.

Utility and usage frequency: Lindlbauer et al. [20] optimised the virtual content according to its utility and usage frequency, to adapt how much information to show and where to place it.

Semantics: Cheng et al. [5] proposed an optimisation based on the semantic meaning of the environment. Qian et al. [34] presented an authoring tool for designers to create semantic associations between virtual objects and the real environment, e.g. to place a pdf document next to a notebook.

However, these approaches are limited in two aspects. First, most of these approaches lack any form of user control over the resulting placement optimisation, while adding interaction into optimisation systems (i.e. human-in-the-loop) has been shown to be beneficial and appreciated by users in other contexts [21]. This refinement enables users to refine the model to produce desired or acceptable organizations, and can be based on intimate, personal or subjective preferences which cannot be inferred computationally. Second, for the most part, these constraints have been designed and tested in isolation from each other, leaving aside the question of how to allow the user to define several constraints in a fluid and natural way. We thus lack a holistic approach that considers and unifies both the various constraints and the input interaction, to bring interactivity into augmented reality layout optimisation systems.

2.4 Using hand gesture to author AR/VR applications

We now need to consider an appropriate input method allowing the user to author the optimization constraints. Hand gestures are interesting candidates as they remove the need for any input device, they offer multiple degrees of freedom and they can provide semantic meaning through different gestures. Following the use of gestures for surface computing [28, 47], hand gestures have already been successfully used for authoring AR and VR applications [33]. For instance, Arora et al. [2] explored the use of mid-air gestures to author animations in virtual reality. They derive a set of design guidelines for gestural animation using mid-air gestures, such as direct manipulation, which enables complex motions in space and the adoption of a coarse-to-fine workflow. Wang et al. [46] propose an authoring tool, GesturAR, supporting users for creating in-situ freehand AR applications. Their tool is based on embodied demonstration and visual programming to create both static and dynamic gestures. Authors demonstrate the usefulness and usability of these gestures for different scenarios, as, for instance, the creation of interactive objects. Yan et al. [49] demonstrate that object-gesture mappings in virtual reality are highly intuitive, favoring gesture discoverability and memorization.

Our proposition is thus built on these previous contributions and it explores the use of hand gestures for authoring user-defined constraints in augmented reality environments.

3 USAGE SCENARIO

Before describing the design space, we first illustrate usage scenarios of the proposed user-driven constraints system. We envisage a use case scenario of a daily life at work of a researcher, John. John's workspace consists of a combination of physical and virtual objects. Physical objects include a desk, laptop, mouse, as well as decorative elements and surrounding walls. Virtual objects exist in augmented reality and consist of virtual notes, virtual documents, as well as virtual tools and widgets.

After a busy week, John's workspace has become severely unorganised with virtual content scattered on his desk and walls. Tidying up these virtual objects manually would be tedious. He would like the ability to easily arrange the virtual objects but within a certain level of constraints and flexibility. John decides to use the user-driven constraints layout optimisation tool.

First, he wants the notes scattered on the wall to be aligned near the corner. However, he does not want these notes to cover his best paper award certificate. To achieve this layout, he first created an *exclusion surface* covering the certificate, followed by an *attractive edge* along the corner of the wall. The edge causes the notes to move toward the corner while the exclusion surface prevents notes from covering the certificate.

Next, he would prefer to see his non-work-related to-do list on the surface in front of him, but only after closing the laptop lid to prevent being disturbed by work unrelated activities during working hours. To achieve this occlusion-aware layout, he created an *in-view surface* right behind the laptop. When John next opens his laptop lid, hiding this surface region from view, any notes placed there will be automatically moved elsewhere.

Now, there is only one more thing to do. John likes listening to music and wanted to have easy access to a music player. He then created another attractive edge on the edge of the desk. This time, he defined a *semantic attractive* edge that only attracts a specific type of content, which he assigns to the music player. Figure 1 illustrates the final state of John's workspace.
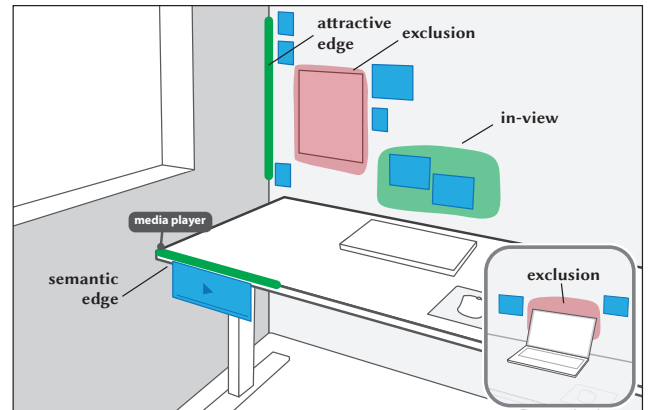

Figure 1: An illustration of a simple usage scenario involving attractive edge, exclusion surface, in-view surface, and semantic attractive edge. The mini figure on the bottom right illustrates the behaviour of the in-view surface when the laptop occludes the virtual content. Virtual content is illustrated in blue throughout the paper.

There are cases when it may be useful for John to define a *preference* for multiple containment surfaces. A containment surface is a user-defined 2D region that attracts virtual content. For example,

John knows he will need quick access to an email list and notes. For this, he creates a high priority containment surface near his mouse pad. John also wants to have an additional surface on which he can situate virtual research papers downloaded from the internet. For this, he creates a low priority containment surface on the left side of the table. The two containment surfaces now have different priorities assigned (Figure 2-a). After several new virtual objects are instantiated, i.e. new research papers John downloaded, the first few papers are attracted by the high priority surface on the right. Only the last virtual paper downloaded is attracted by the low priority surface on the left, after the high priority surface has reached its full capacity (Figure 2-b-c).
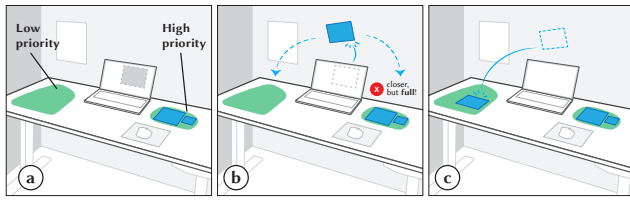

Figure 2: The expect behaviour of preference surfaces according to different surface priority.

We also envisage how our user-driven constraint approach helps in collaborative scenarios, in particular, preparing a shared work space before collaborative activities take place. John has scheduled a quick meeting with his colleague, Sarah. He knows from experience that he will be sitting in his chair, while Sarah will be standing next to him. In this case, John and Sarah will have different *user perspectives* of their environment, and defining surfaces on which they can place virtual content beforehand can help avoid confusion caused by occlusions. Before the meeting, John creates a camera frustum for his own expected perspective and a second one for Sarah's expected perspective (Figure 3-a-b). The intersection between the two surfaces created by these frustums then becomes the shared containment surface used later (Figure 3-c). Items will only be placed in areas visible to both John and Sarah and avoid areas hidden from anyone's view, for instance the space behind John's desktop monitor.
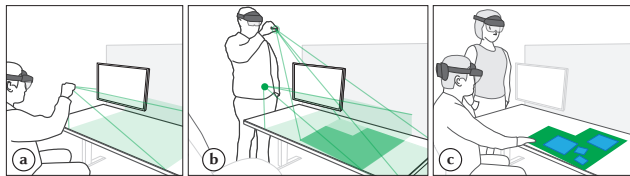

Figure 3: Multiple user perspectives can be authored before collaborative activities to define common containment visible to every user.

4 DESIGN SPACE

In this section we describe how we derive the previously presented constraints into a design framework for user-driven interaction.

Our design space considers three dimensions: the user-driven constraints, the regions of interest, and the constraint parameters. This design space was defined through an iterative design thinking process conducted by three senior and one junior researchers in immersive visualisation, human-computer interaction and mixed reality. We sketched each constraint and used the sketches to drive our discussions. Those sketches were later refined through low-fidelity prototypes in Unity and HoloLens.

4.1 Design objectives

We drive our approach using a set of design objectives based on recommendations from previous works:

- Human in the loop : the driving motivation for our approach is to allow users to refine the virtual content placement optimisation and go beyond the "black box" approach that does not support user interaction. The user can bring new knowledge into the placement approach (e.g. based on personal preferences or unforeseen situations), which brings trust and confidence in the final solution [21].
- Holistic design: our goal is to unify the interaction experience by considering all the interactions of the system and the constraints at once [38]. This leads to more fluid interactions [6] and minimizes the presence of delimiter gestures between the different interactive commands.
- Natural hand gestures: our goal is to adopt a device-less approach based on hand gestures, which are always available. As underlined in section 2.4, hand gestures are highly intuitive and can favor gesture discoverability and memorization [49].
- Direct manipulation: our approach requires to delimit spatial surfaces or edges in 3D. It should thus support direct manipulation, which has been shown to allow performance of complex motions in AR in the context of authoring animations [2].
- Avoid GUIs: As a result of previous objectives, our design should avoid the use of traditional GUIs, which tend to break the interaction flow and have been shown to perform worse than gestures to active commands in augmented reality [36].

4.2 User-driven constraints

We revisited the constraints presented in the related work section from a user perspective, leading to a set of user-driven constraints, i.e. constraints that can be defined or parameterized by the user. One important consideration when deciding which constraints should be user-driven, was to avoid user burden by having the user define constraints that could be easily and efficiently performed automatically. For instance, asking the user to manually tag or classify the environment geometry would be tedious. However letting the user define a specific preferred surface or edge on the environment as a container for virtual content can allow him to personalize the environment.

Our set of constraints focuses on those being related to the real world environment, leaving aside the constraints dealing with the inner arrangement of the content (e.g. window alignment or layout grids). We defined 8 different user-driven constraints, illustrated in Figure 4 - Left. We also described the effect of optimisation for each

constraint. Our constraints are defined as cost functions and are optimised by minimizing the cost. The cost function can consider different dynamic parameters, such as a region's weight or the distance between the content and the region.

4.2.1 Attractive Edge. This constraint allows the user to define an edge near which the content should be displayed. The effect of this constraint on content follows a spring metaphor: if the content moves away from the region of interest, it will be pulled back. From a systems perspective, when the distance between the content and the region of interest increases, the associated cost increases.

4.2.2 Repulsive Edge. This constraint defines the opposite behaviour to the attractive edge. If the content is moved towards the region of interest, the metaphorical spring will push it back. From a systems perspective, when the distance between the content and the region of interest decreases, the associated cost increases.

4.2.3 Containment. This constraint allows the user to define a region that will contain the virtual content. For instance, the user may want to define the surface of a wall or part of a wall as a container. From a systems perspective, when the content position is outside the containment region, the associated cost increases.

4.2.4 Exclusion. This constraint defines the opposite behaviour to containment, i.e. the user defines a region where the content should not be displayed. For instance, the user may want a physical whiteboard not to be occluded by the virtual content. From a systems perspective, when content enters the exclusion region, the associated cost increases.

4.2.5 In-view. This constraint allows the definition of dynamic optimisation over certain regions, which are enabled only when they are in the field of view of the user. For instance, the user may want to assign an in-view constraint to a large surface so that the content of smaller nearby surfaces is arranged on the large surface when the user is in front of it. From a systems perspective, when the user looks at in-view regions, the associated cost decreases.

4.2.6 Preference. The preference constraint allows the user to define the priority order in which containers will be used: when one region is filled with content, the next preferred region will be used to place the content. From a systems perspective, the cost correlates to the priority of the preference regions (e.g. high priority regions will have a low cost).

4.2.7 User Perspective. This constraint allows planning for commonly used viewpoints, or for collaborative activities by anticipating the intended user perspectives from the participants. The user can define one or more static perspectives beforehand, each from a specific position and facing direction. The surface regions in the environment that are visible from all views will automatically be selected as containers. From a system perspective, the surface areas where the frustums from these pre-defined views overlap are used to define a set of containment regions.

4.2.8 Semantics. This constraint allows the user to associate semantics to regions of interest. While some semantic information can be extracted automatically by using computer vision approaches [5], having a user-driven constraint allows the user to define personal semantics. Automatic semantic association has already been demonstrated in previous works and we thus focus on user-defined semantics only. For instance, the user may want to associate a virtual calendar to a position on a wall where they previously hung a physical calendar. From a systems perspectives, when a region has the same label as the content label, the associated cost decreases.

4.3 Regions of interest

We consider four major regions of interest: either a point, a 1D line, a 2D surface or a 3D volume. Each of these regions represents common parts of the spatial environment to which the virtual content may be associated: a position in mid-air (point), the edge along a piece of furniture (1D Line), the surface of a wall or table (2D surface), or a 3D area around a physical object (3D volume).

4.4 Constraint parameters

Some of the constraints we discussed previously depend on particular parameters. For instance, the edge constraint can be parameterized by defining a positive or negative weight to increase the attraction or repulsion of its spring. The edge constraint can also include a minimum distance parameter, which defines the minimum distance from the edge at which content may be placed, leaving a buffer region in between. The list of possible parameters are detailed in Figure 4 - Right.

Our final design space results from the most frequent or relevant combinations of user-driven constraints, regions of interest and constraint parameters, and is illustrated in Figure 4 - Right.

5 GESTURE ELICITATION STUDY

We conducted an elicitation study to explore the design space of the gestures that could be used to define our user-driven constraints, and inform our following system development. In elicitation studies [3, 13, 44, 45], participants are presented with a *referent*, which is imagined to be the effect of the action caused by a gestural *sign*, which the user is asked to generate. The aim is to identify gestures that are intuitive and easily discoverable by users. We asked participants to suggest hand gestures (the signs) for fourteen different referents (constraints).

5.1 Overview and rationale

As discussed in the introduction, we wanted to move away from the inherited UI interfaces that populate current augmented reality platforms that rely on 'legacy' GUI components such as contextual menus and UI widgets, as these break the interaction flow and distract the user's attention from the surrounding world. Instead, our goal was to explore the use of spatial gestures to define the user-driven constraints. Such gestures can ensure a fluid interaction [6].

Our study focuses only on hand gestures, i.e. participants could use one or two hands, or a combination of hands and any of their fingers, as well as pinch and tap gestures. The main reason for this was that we wanted our approach to be feasible with current state-of-the art HMDs, which offer effective hand and finger tracking.

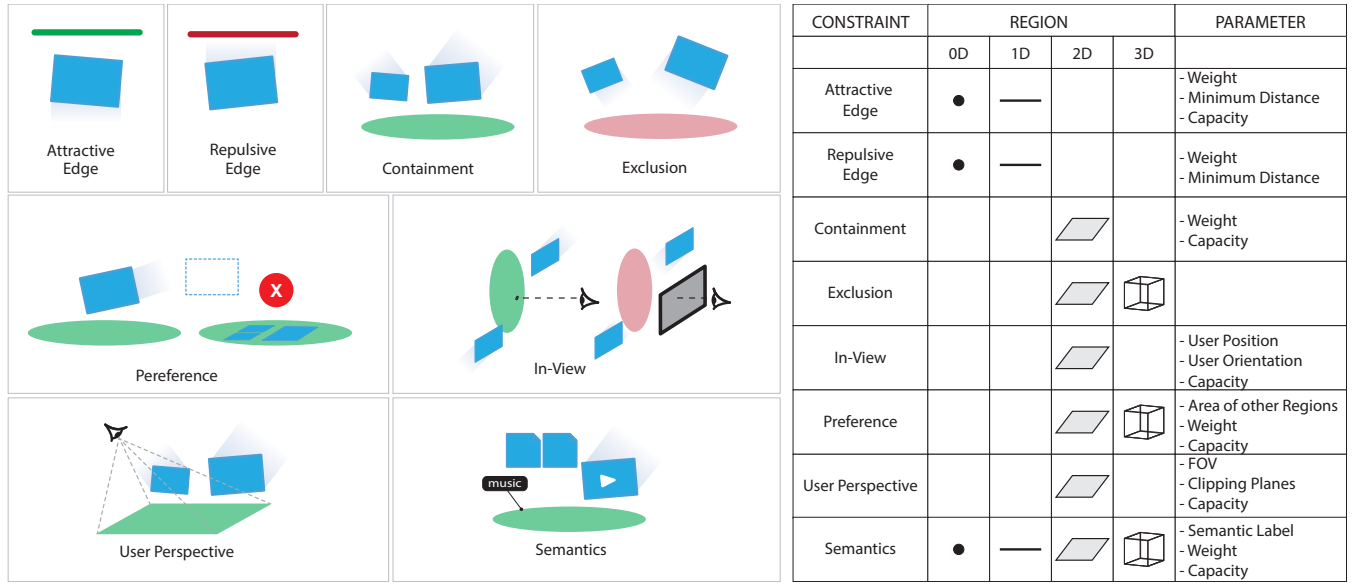| CONSTRAINT	REGION				PARAMETER
	0D	1D	2D	3D	
Attractive Edge	●	—			- Weight - Minimum Distance - Capacity
Repulsive Edge	●	—			- Weight - Minimum Distance
Containment			▱		- Weight - Capacity
Exclusion			▱	⬚	
In-View			▱		- User Position - User Orientation - Capacity
Preference			▱	⬚	- Area of other Regions - Weight - Capacity
User Perspective			▱		- FOV - Clipping Planes - Capacity
Semantics	●	—	▱	⬚	- Semantic Label - Weight - Capacity

Figure 4: Left: Our user-driven constraints, illustrated on either an edge or a surface region. The blue window represents the virtual content whose placement is optimised. Right: Our design space for the user-driven specification of dynamic constraints.

5.2 Referents

In our study, we consider a referent as a combination of a user-driven constraint, a region of interest, and (optionally) a specific parameter.

We asked participants to create signs for the eight constraints of our design space: attractive and repulsive edge, containment, exclusion, in-view, preference, user perspective and semantics. Each constraint was coupled with between one to three regions and zero to two parameters (see Figure 4). We chose a total of 14 referents, including a command to remove the created constraints. While many more interesting referent are possible, we limited the study to 14 referents to keep the study length under one hour. We asked participants to propose up to three gestural signs for each referent and choose their preferred one. Requiring users to produce multiple interaction proposals for each referent, a technique known as Production, can reduce legacy bias in user elicitation studies [27]. Other techniques to reduce legacy bias such as Priming and Partners [27] were less well suited to our study: users were already in a novel environment, hence priming users to think with new forms of interaction could be confusing, and working in groups was difficult while wearing the headset. Priming also presents a risk of unintentionally influencing or constraining the participant suggestions.

5.3 Participants

Fourteen students (7 females, 7 males) volunteered for our study. Their average age was 28.07 (SD = 3.02). All participants were right-handed. Twelve participants were PhD students, one was a Master Degree student and one is a Postdoctoral researcher. Six participants had prior knowledge of AR/VR systems.

5.4 Apparatus

The study was conducted on a Hololens2 wirelessly connected to a laptop using "Holographic Remoting" in Unity. This allowed holographic content to be streamed to the Hololens in real time from play mode in Unity, and allowed us to control the study on the laptop and reduce the computational load on the Hololens. When participants wore the HMD, their hands were augmented by virtual hand joints that were detected and displayed by the Hololens to provide them with feedback. For instance while performing a raycast, the system highlighted the spherical index finger joint. We wrote a C# script for Unity to log participants' gestures on button press from the keyboard. The Unity application did not recognize user gestures, it only tracked and recorded the participants' hands while preforming a gesture. Once the gesture was finished, the system saved the recording of the position and orientation of the head and all hand joints from both hands to a CSV file. The timestamp of each object was also recorded to allow to synchronise hands and head during results analysis.

5.5 Design

Our study followed a within-subject design with one factor, the Referent (14 possibles values). We used a Latin Square to counterbalance the order of the referents.

5.6 Procedure and setup

The participants were seated at a desk in front of a wall. Such setup allows for gesture creation in horizontal and vertical surfaces, as well as in mid-air. For participants with no prior knowledge on AR/VR systems, we familiarised them with the default gesture recognition provided by MRTK and let them see how hands are tracked.

Once the participants were familiar with their task and comfortable wearing the Hololens2, we orally described the example of each constraint behaviour and its effect on the content, without hinting at any possible user interactions to avoid biases [35]. We used the illustrations of Figure 4 to help participants understand the constraints, as these images do not afford any interaction. We gave participants time to think and describe the gesture they were to make. Then, we took notes based on our observations and their feedback, and recorded their gestures. Participants were given a $20 gift cards for their participation.

5.7 Methodology

To analyse the results of the elicitation study, we used the CSV data to replay the gestures in Unity after the study and code them. A first coder proceeded to create an initial gesture classification that was refined with another coder. Once all gestures coded, we measured the agreement rate (AR_i) for each referent. It was calculated using Equation 1 proposed by Tsandilas [44]. Having a closer look at Tsandilas [44] notation, q is the total number of unique signs (i.e. gestures) produced, n_{ik} is the number of occurrences of a sign for referent R_i, and n_i is the total number of signs suggested for referent R_i. The overall agreement rate AR is the average of all AR_i.

$$AR_i = \sum_{k=1}^{q} \frac{n_{ik}(n_{ik} - 1)}{n_i(n_i - 1)} \quad (1)$$

5.8 Results

We collected for each participant one preferred gesture for each of the fourteen referents, leading to a total of 196 collected gestural signs for the study.

5.8.1 Gesture Categorisation. When analysing the results, we discovered that similar gestures had been suggested for different referents: these gestures differed in details such as the hand used (dominant or non-dominant), the number of fingers, or the combination of simple gestures (tap or pinch). To categorize the collected gestures, we decided to consider two gestures identical if they only differed on the finger used. For example, some participants started their gesture with a pinch using the thumb and index fingers, whereas others did it with the thumb and middle fingers.

5.8.2 Gesture agreement and preferred gestures per referent. The average agreement rate for all referents was 0.26. We illustrate the agreement rate for each referent in Figure 5. The figure is color coded according to the classification of Vatavu and Wobbrock [45] for agreement rate values: low (yellow, < 0.1), medium (blue, 0.1 - 0.3), high (light green, 0.3 - 0.5) and very high (dark green, >0.5). None of the agreement rates we observed fell into the 'low' category. We hereafter describe the most common gestures for each constraint.

Attractive Edge: The most common gesture to create an attractive edge was to do a pinch with the right hand and drag it along where the edge should be positioned. The second most common gesture was to do a pinch with both hands and drag inwards. The attractive edge constraint could be defined with one of two parameters: the minimum distance and the weight.
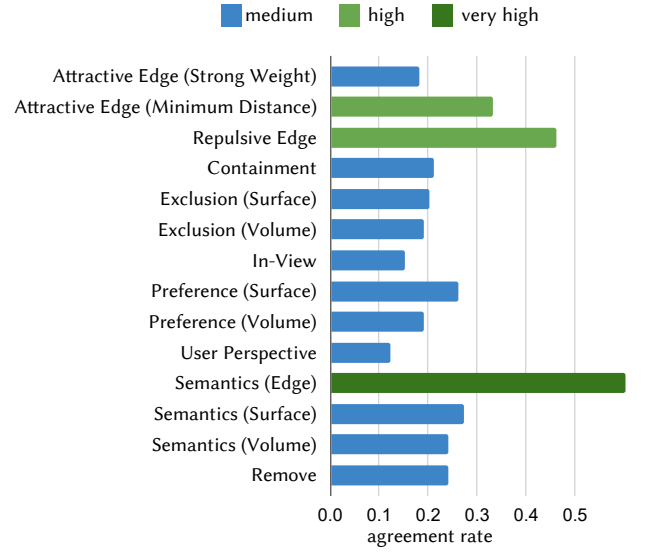

Figure 5: Agreement Rates for each referent. Results are color coded according to the classification of Vatavu and Wobbrock [45] for agreement rate values.

To define a minimum distance between a virtual object and an attractive edge, the most preferred gesture was to indicate the distance perpendicular to the edge by adding a drag or pinch gesture. The second most common gesture was to use the distance between thumb and index finger after releasing a pinch.

To define an edge with a strong weight the most common gesture was holding and waiting after creating the edge. The second most used approach was to add another gesture after edge creation, such as a pinch or a tap.

Repulsive Edge: To create a repulsive edge, participants tended to use the opposite gesture from the attractive edge. The most preferred gesture was to do a two-handed pinch and drag outwards, and the second one was the same gesture with a single hand.

Containment: The most common gesture to define a container region was to perform a freeform raycast with the hand around the region. Others imagined a gesture similar to the menu invocation on HoloLens, i.e. opening all fingers at the same time.

Exclusion: To create an exclusion region, most participants performed the same gesture as for containment, but extended it with an additional gesture, such as double tap. The second most preferred gesture was scribbling in front of the real surface. This result is interesting, as one could expect that participants would propose the opposite gesture from containment.

In-view: To create an In-View region, most participants performed a freeform raycast region outline using both hands. The second most common gesture was representing the user's view: for instance, pinching with two hands while moving them down or towards the user.

Preference: To tag the preferred regions, participants adopted the same gesture as for *Containment*, but using the non-dominant hand. The second most preferred gesture was to create anchor points defining the outline of the preference region.
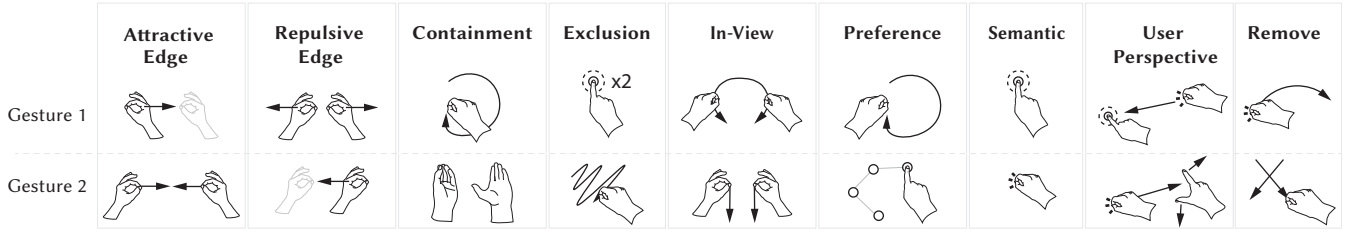
Figure 6: Gesture 1 and Gesture 2 are the most frequently proposed gestures for each constraint.

User Perspective: Participants proposed to define the user's field of view (FoV) by pinching at the origin of the FoV and dragging towards the direction of the frustum. The second preferred gesture was to simulate an eye blinking by extending the index and thumb fingers, orienting the hand in the direction of the frustum.

Semantics: Participants proposed to double tap or pinch at a created region to attach the semantics label.

Removing constraints: Finally, when asked which gesture to use to remove the created constraints, participants proposed doing either a crossing gesture or a cross mark on the constrained region.

5.9 Summary of findings

Regarding the gestures themselves, we were surprised to see that the collected gesture set was made of variations of simple gestures, rather than more complex, semantic-oriented or mnemonic gestures (e.g. drawing a letter in mid-air). It is also interesting to note that the selection of the regions was almost always performed using raycasting. Many gestures started with the raycasting gesture and were followed by a specific simple but meaningful gesture to define the appropriate constraint.

This highlights that most participants adopted an Object-Action interaction model [41]: in this model, the user selects the object first (the region of interests in our case) and then selects the action which will be performed on the model (the user-driven constraint in our case). As this model became prevalent in Graphical User Interfaces (replacing the prior Action-Object model used with command prompts), it has since been adopted in recent implementations of augmented reality interfaces, for instance with the appearance of a mid-air context menu following an object selection. Hence the results of our study are in line with these recent demonstrations of the Object-Action approach for augmented reality.

6 SYSTEM DESIGN AND IMPLEMENTATION

We designed and implemented a set of gestures and their underlying optimisation for user-driven constraints. To follow our design objectives, we propose a uniform set of gesture that limits the need for delimiters and favours interaction fluidity.

6.1 Overview of the user operations

The user-driven optimization mode can be started through a system menu or shortcut. Then the user can perform the different gestures without the need for any delimiter, following the operations defined in Figure 8: all gestures begin from one of the four initial states (pinching with two hands to create an edge, pinching with one hand to create a surface, pinching while moving the hand or clicking on

a region). Each constraint creation results from following a unique path of actions from one of these four states. The user can also delete a constraint by using a specific gesture.

Regarding the optimization of the content layout, there are two possible options: either separating the constraints creation from the content layout optimization, or doing both at the same time. Our informal tests revealed that the first option was not adequate, as participants were not sure where the content would go after creating several constraints. Instead, we adopt the second approach, i.e. the surrounding content position is dynamically optimized as the user creates the constraints. The constraint surfaces and edges are visible during the entire operation. Finally, once the user is satisfied with the virtual content arrangement, the user-driven-optimization mode can be stopped again using the same initial shortcut or system menu.

6.2 Final set of gestures

6.2.1 Edge-based constraints. While in our study participants used one or two hands to create edges with constraints, in our final gesture set we decided to use the two hands for both constraints to be consistent. The *Attractive edge* is created by pinching and moving both hands inwards, whereas the *Repulsive edge* is defined by moving them outwards, as illustrated in Figure 7top. The length of the edge on which the attraction/repulsion applies is constant and defined by the distance between the two pinches. Once the edge is created, the user can add weight and minimum distance parameters to it, without releasing the edge. To add a weight parameter, the user can stretch the edge. The thickness of the edge changes according to the weight to provide visual feedback. To add a minimum distance parameter, the user can move the right hand tangentially to the edge.

6.2.2 Surface-based constraints. In our study, for some of the constraints (containment, exclusion, in-view, preference and semantics), participants often started by performing a common gesture to define the surface region, followed by an additional gesture. In our design, we decided to group these constraints into the same gesture state machine (see Figure 7).

We illustrate the proposed gestures for the case of a simple container. First, the user creates the surface region through a raycast. If no other gesture follows the surface creation within a time threshold, this surface is considered to be a *Containment surface*. To define the other constraints, the user can perform different gestures within the given time threshold after creating the surface. A scribble gesture inside the surface will define an *Exclusion Surface*. A
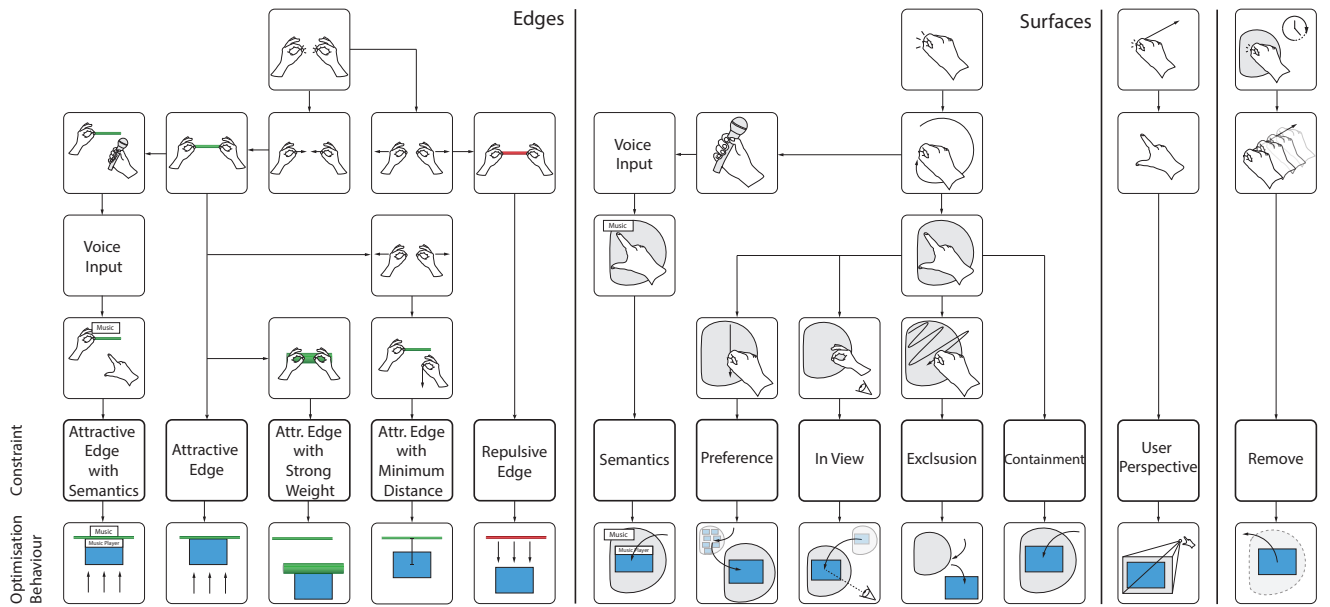
Figure 7: Complete diagram describing the final set of gestures of our system. All gestures begin from one of the four states shown at the top, each following a unique path to the constraints at the bottom.

pinch gesture in front of the surface while looking at it will define an *In-View Surface*. A drag down gesture will define a *Preference Surface.*

To create a *Semantic Surface*, we decided to use voice input to define the semantic label. We adopted this voice-based input to avoid the use of any keyboard virtual widget and follow our initial design objectives. To activate voice input, the user gathers the fingers of the right hand together while holding the pinch, as if holding a microphone.

6.2.3 User perspective. To create the fixed viewing perspective the user performs a pinch gesture to define the view direction, then drag it at the planned center of the view frustum.

6.2.4 Region removal. The last gesture is to remove any of the created constraints. First the user can select a constrained region by pinching and holding it. Then he can throw it away or drag outside of the field of view to remove it.

6.3 Implementation

We implemented a prototype integrating our gesture set and the optimisation approach using MRTK for Hololens 2 and Unity.

6.3.1 Gesture. We implemented the gestures with the MRTK core services input system, to detect hands and to get data from pointers. Once the environment is registered by the MRTK spatial awareness system, we used the integrated raycast pointer to perform interaction. For example, a surface container is created by pinching and dragging the raycast pointer on the environment mesh. When the system detects a drag gesture, the system saves the pointer position into a list. The system provides the user with visual feedback of the trace of the raycast, viewed through the AR display. When releasing

the pinch, the system iterates over the list of previously created points and generates a container mesh. Once the container created, we empirically defined a threshold value of 2 seconds to allow the user to perform an additional gesture to change the container type. These following gestures (see Figure 7) were implemented using the MRTK built-in capabilities.

To create an attractive or repulsive edge, once both pinches are detected, the system waits for inwards or outwards movement. We empirically defined a distance threshold of 0.03 m. If the distance between the initial and final pinch positions exceeds the threshold in the inwards direction, the edge type is considered attractive. Conversely if the threshold is exceed in the outwards direction, the edge type is repulsive. We also empirically introduce another threshold of 0.04m. If the distance from initial pinch position exceeds it (always in the outward direction), then the weight of the edge is changed in proportion to the distance moved. The cylinder width is enlarged accordingly for visual feedback.

6.3.2 Optimisation. Once the region with a specific parameter is created, we optimise our virtual content placement using the MRTK SolverHandler. Every virtual object has a Solver script attached to it. To attach the virtual content to a given region, we adapted the MRTK solver SurfaceMagnetism to our needs. By default, the SurfaceMagnetism allows virtual objects to be attached to surfaces when looking or pointing at them, as well as to the custom surfaces. Since we wanted to specify the region where to attach our virtual objects, we added weights to each type of region and changed them dynamically depending on the type of constraint applied to the region. The system then searches for the highest region weight and uses it to attach virtual objects. Another default behaviour of SurfaceMagnetism we wanted to change is virtual object attachment

(a) attractive edge

(b) repulsive edge

(c) containment

(d) exclusion
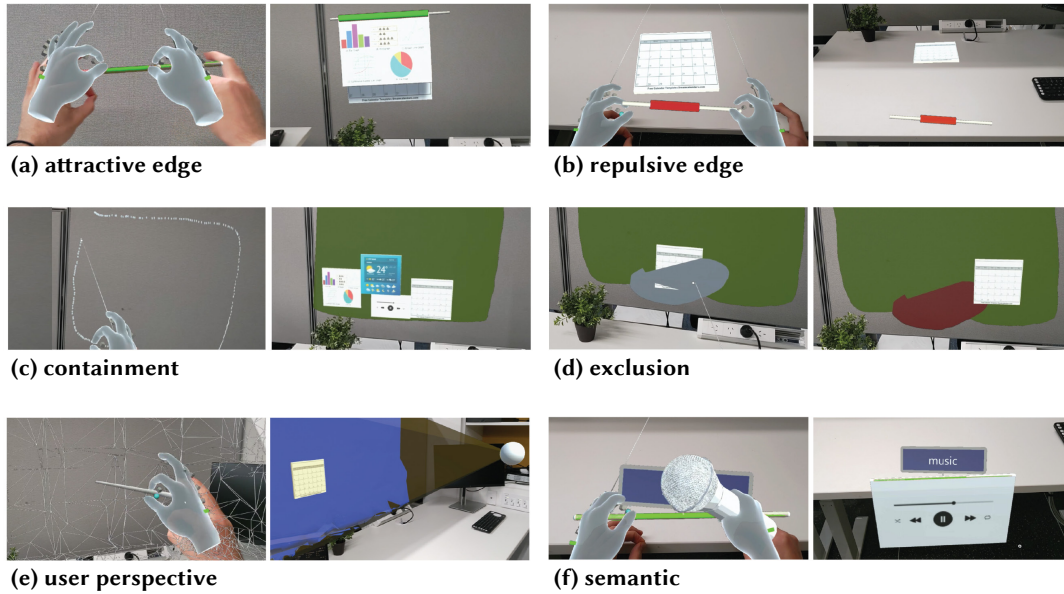
(e) user perspective

(f) semantic

Figure 8: Illustration of the implementation of some of the user-defined constraints. For each constraint, we illustrate the gesture (on the left) and the resulting placement optimisation (on the right).

to the center of the custom Transform. In case of multiple virtual objects, they all will be attached to the center of the container. Thus, we constrain the virtual object's movement towards center of the container once it is inside. In order to keep the virtual content visible and not overlapping, we used collision detection handled by Unity Physics Engine.

To optimise virtual content position for an attractive edge, we first create an edge object and attach virtual content to it. To apply proper orientation of the virtual content, we use Unity Physics Engine to create a hinge connecting the content to the edge. Once this is done, we apply and tweak the MRTK Follow Solver to the edge. To implement the behaviour of the repulsive edge, we also use the MRTK Follow Solver and use the Minimum Distance parameter as a repulsion distance. We apply repulsion only in a direction parallel to the edge forward vector.

6.3.3 Optimisation when combining constraints. When combining different constraints, we have to address the question of how to associate the virtual content already present in the environment to each constraint. We adopt a distance-based approach, where each constraint (except semantics) has an area of effect defined by a distance threshold. The semantic regions attract virtual content wherever it is located. The other virtual objects within this threshold are affected by the constrained optimization. This provides the user with the flexibility of associating virtual content to a given constraint by simply approaching the window to the constraint surface/edge, or to redirect content between constraints if desired.

6.3.4 Constraint capacity. Each constraint, either surface or edge, has a limited capacity: a small containment area (i.e. smaller than a widget) can contain only one single widget, and the widget spans out of the area. Other widgets that overpass the capacity of the

containment area are not optimized (i.e. do not move from their location). If the user defines a preference surface, those widgets that cannot fit move to the preference surface.

7 SUMMATIVE STUDY

The goal of this study was to validate the use of our gestures for creating a user-driven layout optimisation. Another goal of the study was to see how much of the usage scenario mentioned earlier could be achieved with the current gestures and implementation. In particular, we wanted to know how useful the semi-automated layout approach is for content arrangement tasks compared to the manual adjustment approach.

7.1 Study design

7.1.1 Tasks and Instructions. In this study, we asked participants to create an augmented reality layout using two different methods, i.e. manual placement and user-driven constraints. The participants were provided with 9 virtual contents in front of them: four post-it notes, one calendar, one graph, one music player, one playlist and one weather widget. First, we asked participants to manually place the content in an initial position as a warm-up activity, to get them used to the environment (see Figure 9-left).

Then they were asked to adjust the position of these widgets in two steps, either using a manual approach or with our user-driven constraints. The content was initially placed nearby the area where it should be moved to, so as not to hinder the manual condition. When using the user-driven constraints, each step involved a different set of constraints (see Figure 9- center and right). The first step consisted in creating an attractive edge on the table, a semantic edge on the top of the side wall, an exclusion surface over a poster hanging on the front wall, a user-perspective surface on the

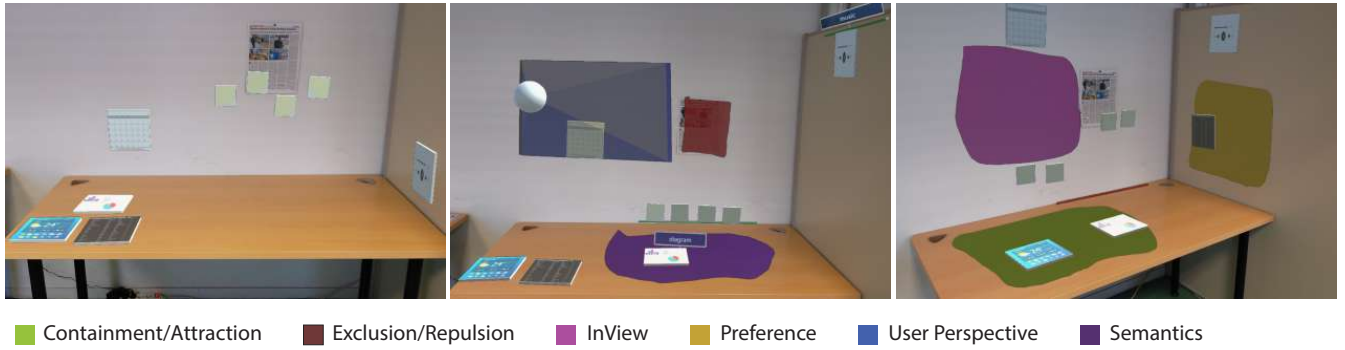| Containment/Attraction | Exclusion/Repulsion | InView | Preference | User Perspective | Semantics |

Figure 9: Our user study started with the virtual content manually distributed in the space during the warm-up activity (left). Then users had to arrange the content in two steps, either using a manual approach or with our user-drive constraints. With our approach, the first step (center) involved Exclusion, Attractive Edge, Semantic Edge, Semantic Surface and User Perspective constraints; the second step (right) involved Repulsive edge, Preference, Containment and In-View constraints.

front wall and a semantic surface on the table. Then participants removed all constraints before moving to the next step. The second step consisted in creating a repulsive edge on the same position as the previous attractive edge, an in-view surface on the front wall, a preference surface on the side wall and a container surface on the table. We decided to decompose the study into two steps to make each step meaningful to the participants (i.e. we explained the reasons to readjust the content before each step) and because the use of all the constraints at once would have required a larger environment.

7.1.2 Techniques. We considered two conditions: our user-driven constraints, and a baseline consisting of manual content placement. Our comparison does not provide a complete comparison of all possible configurations, however, provides an initial qualitative feedback about the feasibility of our gesture-based approach versus a naive approach. We decided to leave aside the comparison with a fully automatic approach, for two reasons: first, such an approach would not have the same capabilities than our system (i.e. let the user define his layout); second, there is no system integrating all existing layout optimisation approaches, hence developing such a system poses important design and development challenges.

7.1.3 Setup and apparatus. Our setup replicates the initial scenario of our paper: we conducted the experiment around a table located in the corner of the room. This setup allows us to create surfaces and edges on three different planes. Participants were wearing a Hololens 2 while standing in front of the table.

7.1.4 Participants. Twelve students (6 females, 6 males) from our local university volunteered for our study. Their average age was 25.3 (SD = 2.9). 10 participants were PhD students and 2 were Master Degree students. 3 participants had prior knowledge of AR/VR systems, and none of them had prior knowledge on constraint-based optimisation.

7.1.5 Study Design and procedure. The study followed a within subject design with the Technique as only factor (Manual, User-Driven Constraints). The study was divided into two blocks, each corresponding to one Technique. We counterbalanced the order of

the Technique among participants. Each block was divided into two steps, where participants had to arrange the content as instructed. Before each step there was a training phase, where we showed participants the gestures to create each constraint and let them try them until they felt confident. We used a tablet to show participants the images of the final virtual content arrangement that they should try to reach using each Technique. They had to arrange the content until they felt it was similar to the illustration. For participants with no prior knowledge on AR/VR systems, we familiarised them with the default gesture recognition provided by MRTK before starting the study. We told participants that they could take a break when wanted. Each session lasted 1h 45 minutes on average.

7.1.6 Data Collection and Analysis. After each Technique, we asked participants to fill a NASA-TLX to measure their perceived workload (mental demand, physical demand, temporal demand, performance, effort, frustration level) on a 100 point scale (lower is better except for Performance), as well as to rank the level of agency (i.e. to what extent the virtual content was placed where they intended [23, 43]). To analyze the NASA-TLX and the Agency results, we performed t-tests.

For the user-driven constraints, we asked participants to provide a feedback after each constraint creation (i.e. gesture) by filling several 7 point Likert scales (intuitiveness, preference, easy to learn, easy to remember, easy to perform, socially acceptable and easy to use). We also asked them to comment on what they liked and disliked about each gesture. At the end of the study, we asked open questions about what could be improved in our system and what other features they would like to use. We also asked participants to rank the two Techniques in order of preference. Since our instructions put the stress on satisfaction rather than on time, we did not record the completion time. We did not measure any error rate either, because all participants had to successfully complete the tasks.

7.2 Results

We first report the results on the gestures, then on the general approach.

7.2.1 Gestures. Figure 10 illustrates the results of the 7-point Likert scales evaluating the gestures for each constraint. Overall, all the gestures had a majority of positive scores for all the evaluated metrics. Interestingly, some of the gestures did not collect any negative score (containment, repulsive edge and exclusion).

All the participants commented that creating the containment surface was very intuitive and simple, confirming the choice for this gesture which is fundamental in our gesture set. P1 commented that "it's the most intuitive gesture and the most used gesture in the environment.". P2 liked it, because of the ability to "*create arbitrary surfaces, not only squares*".

The gestures to create the attractive and repulsive edges were commented to be "*straightforward*" and "*easy to learn*" for P6, P7, P11. To create a repulsive edge, P2 noted that "*the outward movement is intuitive*", and P10 liked that "*the gesture is the opposite of the attractive one*". On the contrary, P1 found it "*confusing to remember the difference between attractive and repulsive edge*". A couple of participants mentioned that they would like to move an edge or change its length after creation (P2, P6), which is not yet available in our current prototype. Some participants (P2, P6, p9, P11) mentioned that the gesture to create an InView surface requires speed and more precision when clicking in front of the surfaces.

Regarding the exclusion gesture, while some participants commented that "*scribbling is fun*" (P4, P9), P2 noted that "*it is not clear for how long to scribble*" and P6 that "*the transition between the two gestures requires thinking at the beginning*".

The semantics gesture was appreciated in general: P1 liked to use different modalities (voice and gestures) and P6 commented that "*the microphone gesture was easy to remember*". However, P3 and P5 did not like the voice input because they did not want to speak in front of others, as commented by P3: ""*I don't like talking out loud to a computer, seems weird to an outside observer.*"". While the gesture to create a user perspective was one of the most intuitive, some participants commented that they wished to see a preview of the surface before releasing the pinch.

7.2.2 Manual vs User-Driven Constraints. When asked at the end of the study which approach they preferred, all participants indicated a preference for the user-driven constraints over the manual condition. When asked to motivate their preference, participants said that the user-driven constraints are "*faster*" (P3, P5, P6, P8, P12), "*easier*" (P1, P9, P11, P12), "*give more control*" (P4), "*require less precision*" (P7), allow to "*create specific container for specific things (musics, diagram, ...)*" (P1) and "*allow to move many things at the same time*" (P7, P8). P10 commented that although it "*takes time to get used to it, I believe once you do you can efficiently organize everything*". So the main motivation for preferring the user-driven constraints seems to be the efficiency rather than the quality of the resulting layout. This can be explained by the fact that our task was relatively simple, and users could reach the intended layout with both approaches. This was confirmed by the results on the Agency showing there was no significant difference on the perceived level of control over the content placement using both the manual approach (M = 74.58, SD = 15.29) and the user-driven constraints (M = 65.83, SD = 18.80, t = 1.25, p = .11).

The results of the Nasa-TLX (Figure 11) confirm this, as there was a significant increase in Temporal demand for the manual approach (M = 77.08, SD = 18.02) compared to the user-driven constraints (M = 45.83, SD = 19.04, t= 4.13, p = .0002). There was also a significant increase in Physical demand for the manual approach (M = 57.92, SD = 15.44) compared to the user-driven constraints (M = 42.5, SD = 15.22, t = 2.26, p =.01). There was no significant effect of the Technique on the other Nasa-TLX metrics (Mental demand: t = 0.62, p = .27; Performance: t = -0.81, p = .21; Effort: t = 1.39, p = .08; and Frustration: t = 1.59, p = .06).

Participants also commented on the utility of some constraints. Many participants liked the idea of having an exclusion area. P1 noted that: "*The behavior of the constraint is obvious, we know exactly what we are doing*". The alignment resulting from the attractive edge was appreciated, as underlined by P2's comment: "*The idea of straight lines to which objects attach is good*". P11 commented on the user perspective constraint that, "*It's really useful to create an area for a precise perspective I would like to have in the future*".

7.2.3 Improving our system. We also gathered participants feedback about what could be improved and what other features they would like to use. Regarding the system improvement, participants would like to have a better gesture recognition. We noticed that for participants with limited or no experience in AR/VR, it was hard to understand the limitations of the Hololens in terms of gesture detection, such as a reduced FoV and hand recognition inaccuracy. Some participants mentioned that if they performed the 2-step gestures too slow, a containment surface was created instead of the intended one. Another suggestion was to smooth the shape of the surfaces after creating them. In terms of new features, participants would like to be able to define the capacity of the surfaces and size of the region of effect, whose boundaries should be made visible. They would also like to move surfaces and edges after their creation. P8 suggested to let the users define the direction of attraction or repulsion, instead of using the surrounding surfaces.

7.2.4 Summary. This summative study allowed us to collect first feedback on our gestures and on the interest of our approach. Overall the gestures were appreciated and found easy to perform. Our user-driven approach was preferred to a manual arrangement, and requires lower workload on the temporal and physical demands. Besides this study allowed us to test the use of several constraints at the same time. These are promising results, even though they were collected on a controlled use case with a limited number of virtual content. We also gathered valuable feedback that will allow us to improve the system in the future, in particular towards providing users with even more control over the system.

8 DISCUSSION AND FUTURE WORK

8.1 Complexity, legacy and consistency of user-defined gestures

Our first study revealed some criteria that people tend to apply when defining hand gestures for user driven constraints. First, participants tried to reduce the complexity of interaction by using simple gestures, often derived from or combined with other gestures. For example, if the edge with an attractive spring was created by dragging from left to right, the edge with repulsive spring was created from right to left. Second, they wanted the interaction to be consistent for creating regions and constraints. For instance, the

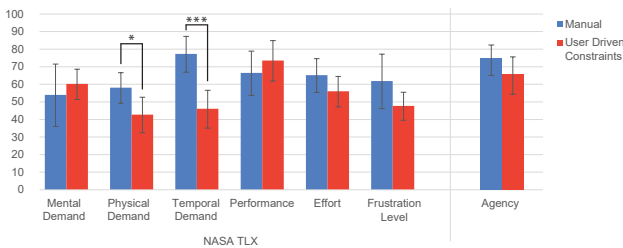**Figure 10: Likert scale results for the gesture ranking**

Figure 11: NASA TLX and Agency rankings for each technique.

semantic label was added with the same gesture for all types of region. Despite our push to move away from inherited UI approaches, participants used an Action-Object approach common in desktop and mobile GUIs in the vast majority of cases. This is coherent with previous works that recommend gestural manipulations enabling users to manipulate virtual objects rather than commands using symbolic gestures [1]. Our elicitation study allowed us to avoid the use of menus or GUI widgets, allowing users to maintain their focus on the physical surroundings.

8.2 Validation of the user-driven optimisation

One of the goals of the summative user study is to validate if the gestures for creating a user-driven layout optimisation are usable and allow to fulfill our initial design objectives. The results show that participants felt a similar level of control with our approach than with a manual interaction, with a higher feeling of efficiency, confirming the interest for bringing interactivity into optimization approaches (Human in the loop objective). Overall, people were able to use the various hand gestures to configure their layout without any explicit gesture delimited (Holistic design objective) or interactive device (Natural hand gestures objective). Creating

a containment surface with a direct hand ray, a fundamental gesture in our design, was clearly appreciated and found intuitive, as well as the bimanual gesture to create an edge (Direct manipulation objective). Overall, participants were able to define a complex spatial layout without the need for any GUI menu (Avoid GUIs objective). Our study also provides a number of improvements that we will investigate in the future, such as providing clear feedback and instructions for the gestures, particularly for novice users. For instance, highlighting the area where the gesture is applied before the pinch is released. One of the future challenges will be to integrate some of the mentioned improvements, such as defining the direction of attraction, while preserving our initial design objectives (i.e. limited gesture delimiters, no UIs). One solution could be to perform touch gestures on the surrounding surfaces [31].

8.3 Virtual content beyond small 2D widgets

In this paper we demonstrate our approach using virtual 2D widgets such as post-it notes, weather, music or calendar widgets. Our rational is that these virtual widgets are numerous, hence their placement is more tedious than if the system is only composed of a single larger view. However the defined constraints are relatively independent from the content and our approach could be extended to other types of virtual UIs, such as 3D virtual objects, larger windows (e.g. website, large visualisation, small multiple visualisations), freeform widgets [42], or even to the combination with content displayed on surrounding screens [32]. Obviously these contents would bring new challenges that need to be addressed in future works. For instance, the placement of 3D content should consider the third dimension, and maybe require to define volumetric constrained volumes (instead of only surfaces). Very large windows may be difficult to fit within user-defined surfaces or conflict with others.

8.4 Gaps between gesture design and technical capabilities

Implementing the gestures to create a region with a particular constraint was not an easy task. Some participants wanted to use gestures not recognised by the system. For example scribbling using the palm, or removing by swiping with the whole hand. Some of them used the gestures very close to the headset, and the hands were not tracked in such proximity. Our final set of gestures only considers gestures that can be performed with a Hololens2, but our summative study revealed that some participants would like to perform gestures beyond the recognition range of the device.

8.5 Future needs for robust and flexible layout optimization

The goal of our currents system implementation is to demonstrate the general approach of user-defined constraints from the perspective of the gestural interaction. To this end, we used a built-in solver, the MRTK Solver. This solver has some limitations though, particularly when dealing with inconsistencies across various constraints, e.g. when a specific content can be attached to two surfaces, which can provoke unwanted jittering of the content. Adopting a distance-based approach, where the constraints only have a limited area of influence, reduces this problem to very particular cases. Still, developing a more robust optimization system will probably require to use an external solver, such as the one developed by Mellado et al. [26], which has already been successfully used for optimizing the placement of 2D widgets on projected interfaces [29]. Future works may address the non-trivial challenge of extending this 2D optimization to 3D, or predicting the impact on user performance [4]. With this future implementation would come the question of the system ceiling, i.e. how many virtual widgets and constraints can be effectively used, both from a user and a systems perspective.

9 CONCLUSION

In this paper we addressed the challenge of how to let users define the constraints to optimize the content placement in augmented reality environments. To this end, we presented a design space for user-driven constraints, defining three dimensions: the user-driven constraints themselves, the region of interest and the constraint parameters. To explore this design space and the gestures that could be used to implement it, we conducted a user elicitation study where we asked participants to propose gestures for each user-driven constraint. Using the results from the user elicitation study, we designed and implemented a complete set of gestures and the corresponding content optimisation to demonstrate our approach. A final controlled user study validated the interest of our user-driven approach as well as the gestures.

ACKNOWLEDGMENTS

REFERENCES

[1] Carmelo Ardito, Maria Francesca Costabile, and Hans-Christian Jetter. 2014. Gestures that people can understand and use. *Journal of Visual Languages & Computing* 25, 5 (2014), 572–576. https://doi.org/10.1016/j.jvlc.2014.07.002

[2] Rahul Arora, Rubaiat Habib Kazi, Danny M. Kaufman, Wilmot Li, and Karan Singh. 2019. MagicalHands: Mid-Air Hand Gestures for Animating in VR. 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, 463–477. https://doi.org/10.1145/3332165.3347942

[3] Christopher Austin, Barrett Ens, Kadek Satriadi, and Bernhard Jenny. 2020. Elicitation study investigating hand and foot gesture interaction for immersive maps in augmented reality. *Cartography and Geographic Information Science* 47 (01 2020), 1–15. https://doi.org/10.1080/15230406.2019.1696232

[4] Florent Cabric, Emmanuel Dubois, and Marcos Serrano. 2021. A Predictive Performance Model for Immersive Interactions in Mixed Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 202–210. https://doi.org/10.1109/ISMAR52148.2021.00035

[5] 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* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. https://doi.org/10.1145/3472749.3474750

[6] Niklas Elmqvist, Andrew Vande Moere, Hans-Christian Jetter, Daniel Cernea, Harald Reiterer, and TJ Jankun-Kelly. 2011. Fluid interaction for information visualization. *Information Visualization* 10 (10 2011), 327–340. https://doi.org/10.1177/1473871611413180

[7] Barrett Ens, Benjamin Bach, Maxime Cordeil, Ulrich Engelke, Marcos Serrano, Wesley Willett, Arnaud Prouzeau, Christoph Anthes, Wolfgang Büschel, Cody Dunne, Tim Dwyer, Jens Grubert, Jason H. Haga, Nurit Kirshenbaum, Dylan Kobayashi, Tica Lin, Monsurat Olaosebikan, Fabian Pointecker, David Saffo, Nazmus Saquib, Dieter Schmalstieg, Danielle Albers Szafir, Matt Whitlock, and Yalong Yang. 2021. Grand Challenges in Immersive Analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 459, 17 pages. https://doi.org/10.1145/3411764.3446866

[8] Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. 2014. Ethereal Planes: A Design Framework for 2D Information Spaces in 3D Mixed Reality Environments. In *Proceedings of the 2nd symposium on Spatial user interactions* (Honolulu, USA) *(SUI '14)*. ACM, New York, NY, USA.

[9] Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. 2015. Spatial Constancy of Surface-Embedded Layouts across Multiple Environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (Los Angeles, California, USA) *(SUI '15)*. Association for Computing Machinery, New York, NY, USA, 65–68. https://doi.org/10.1145/2788940.2788954

[10] Barrett M. Ens, Rory Finnegan, and Pourang P. Irani. 2014. The Personal Cockpit: A Spatial Interface for Effective Task Switching on Head-Worn Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3171–3180. https://doi.org/10.1145/2556288.2557058

[11] 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

[12] 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

[13] Leah Findlater, Ben Lee, and Jacob Wobbrock. 2012. Beyond QWERTY: Augmenting Touch Screen Keyboards with Multi-Touch Gestures for Non-Alphanumeric Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2679–2682. https://doi.org/10.1145/2207676.2208660

[14] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. 2014. FLARE: Fast Layout for Augmented Reality Applications. https://doi.org/10.13140/2.1.3060.4167

[15] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 2015. MultiFi: Multi Fidelity Interaction with Displays On and Around the Body. 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, 3933–3942. https://doi.org/10.1145/2702123.2702331

[16] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-Based Interaction: A Framework for Post-WIMP Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*.

Association for Computing Machinery, New York, NY, USA, 201–210. https://doi.org/10.1145/1357054.1357089

[17] Hans-Christian Jetter, Harald Reiterer, and Florian Geyer. 2014. Blended Interaction: understanding natural human–computer interaction in post-WIMP interactive spaces. *Personal and Ubiquitous Computing* 18, 5 (2014), 1139–1158.

[18] Wallace S. Lages and Doug A. Bowman. 2019. Walking with Adaptive Augmented Reality Workspaces: Design and Usage Patterns. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 356–366. https://doi.org/10.1145/3301275.3302278

[19] Joon Hyub Lee, Donghyeok Ma, Haena Cho, and Seok-Hyung Bae. 2021. Post-Post-It: A Spatial Ideation System in VR for Overcoming Limitations of Physical Post-It Notes. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 300, 7 pages. https://doi.org/10.1145/3411763.3451786

[20] 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

[21] Jie Liu, Tim Dwyer, Guido Tack, Samuel Gratzl, and Kim Marriott. 2020. Supporting the Problem-Solving Loop: Designing Highly Interactive Optimisation Systems. https://doi.org/10.48550/ARXIV.2009.03163

[22] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. 2020. Design and Evaluation of Interactive Small Multiples Data Visualisation in Immersive Spaces. 588–597. https://doi.org/10.1109/VR46266.2020.00081

[23] Feiyu Lu and Yan Xu. 2022. Exploring Spatial UI Transition Mechanisms with Head-Worn Augmented Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 550, 16 pages. https://doi.org/10.1145/3491102.3517723

[24] Paul Lubos, Gerd Bruder, Oscar Ariza, and Frank Steinicke. 2016. Touching the Sphere: Leveraging Joint-Centered Kinespheres for Spatial User Interaction. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (Tokyo, Japan) *(SUI '16)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/2983310.2985753

[25] Weizhou Luo, Anke Lehmann, Hjalmar Widengren, and Raimund Dachselt. 2022. Where Should We Put It? Layout and Placement Strategies of Documents in Augmented Reality for Collaborative Sensemaking. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 627, 16 pages. https://doi.org/10.1145/3491102.3501946

[26] Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loic Barthe. 2017. Constrained Palette-Space Exploration. *ACM Trans. Graph.* 36, 4, Article 60 (jul 2017), 14 pages. https://doi.org/10.1145/3072959.3073650

[27] Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, m. c. schraefel, and Jacob O. Wobbrock. 2014. Reducing Legacy Bias in Gesture Elicitation Studies. *Interactions* 21, 3 (may 2014), 40–45. https://doi.org/10.1145/2591689

[28] Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. 2010. Understanding Users' Preferences for Surface Gestures. In *Proceedings of Graphics Interface 2010* (Ottawa, Ontario, Canada) *(GI '10)*. Canadian Information Processing Society, CAN, 261–268.

[29] Aziz Niyazov, Nicolas Mellado, Loic Barthe, and Marcos Serrano. 2021. Dynamic Decals: Pervasive Freeform Interfaces Using Constrained Deformable Graphical Elements. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 493 (nov 2021), 27 pages. https://doi.org/10.1145/3488538

[30] 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

[31] Brice Parilusyan, Marc Teyssier, Valentin Martinez-Missir, Clément Duhart, and Marcos Serrano. 2022. Sensurfaces: A Novel Approach for Embedded Touch Sensing on Everyday Surfaces. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 67 (jul 2022), 19 pages. https://doi.org/10.1145/3534616

[32] Gary Perelman, Emmanuel Dubois, Alice Probst, and Marcos Serrano. 2022. Visual Transitions around Tabletops in Mixed Reality: Study on a Visual Acquisition Task between Vertical Virtual Displays and Horizontal Tabletops. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 585 (nov 2022), 20 pages. https://doi.org/10.1145/3567738

[33] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-Defined Gestures for Augmented Reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (Paris, France) *(CHI EA '13)*. Association for Computing Machinery, New York, NY, USA, 955–960. https://doi.org/10.1145/2468356.2468527

[34] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 65, 18 pages. https://doi.org/10.1145/3491102.3517665

[35] Roman Rädle, Hans-Christian Jetter, Mario Schreiner, Zhihao Lu, Harald Reiterer, and Yvonne Rogers. 2015. Spatially-Aware or Spatially-Agnostic? Elicitation and Evaluation of User-Defined Cross-Device Interactions. 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, 3913–3922. https://doi.org/10.1145/2702123.2702287

[36] Houssem Saidi, Emmanuel Dubois, and Marcos Serrano. 2021. HoloBar: Rapid Command Execution for Head-Worn AR Exploiting Around the Field-of-View Interaction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 745, 17 pages. https://doi.org/10.1145/3411764.3445255

[37] Kadek Ananta Satriadi, Barrett Ens, Maxime Cordeil, Tobias Czauderna, and Bernhard Jenny. 2020. Maps Around Me: 3D Multiview Layouts in Immersive Spaces. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 201 (nov 2020), 20 pages. https://doi.org/10.1145/3427329

[38] Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Gluey: Developing a Head-Worn Display Interface to Unify the Interaction Experience in Distributed Display Environments. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 161–171. https://doi.org/10.1145/2785830.2785838

[39] Marcos Serrano, Anne Roudaut, and Pourang Irani. 2016. Investigating Text Legibility on Non-Rectangular Displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (Santa Clara, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 498–508. https://doi.org/10.1145/2858036.2858057

[40] Marcos Serrano, Anne Roudaut, and Pourang Irani. 2017. Visual Composition of Graphical Elements on Non-Rectangular Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 4405–4416. https://doi.org/10.1145/3025453.3025677

[41] Ben Shneiderman, Catherine Plaisant, Maxine S Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. 2016. *Designing the user interface: strategies for effective human-computer interaction.* Pearson.

[42] Florine Simon, Anne Roudaut, Pourang Irani, and Marcos Serrano. 2019. Finding Information on Non-Rectangular Interfaces. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 102, 8 pages. https://doi.org/10.1145/3290605.3300332

[43] Adam Tapal, Ela Oren, Reuven Dar, and Baruch Eitam. 2017. The Sense of Agency Scale: A Measure of Consciously Perceived Control over One's Mind, Body, and the Immediate Environment. *Frontiers in Psychology* 8 (2017). https://doi.org/10.3389/fpsyg.2017.01552

[44] Theophanis Tsandilas. 2018. Fallacies of Agreement: A Critical Review of Consensus Assessment Methods for Gesture Elicitation. *ACM Trans. Comput.-Hum. Interact.* 25, 3, Article 18 (jun 2018), 49 pages. https://doi.org/10.1145/3182168

[45] Radu-Daniel Vatavu and Jacob O. Wobbrock. 2015. Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. 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, 1325–1334. https://doi.org/10.1145/2702123.2702223

[46] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 552–567. https://doi.org/10.1145/3472749.3474769

[47] Jacob Wobbrock, Meredith Ringel Morris, and Andy Wilson. 2009. User-Defined Gestures for Surface Computing. In *CHI '09 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1083–1092. https://www.microsoft.com/en-us/research/publication/user-defined-gestures-for-surface-computing/ Best Paper Nominee.

[48] Xuhai Xu, Alexandru Dancu, Pattie Maes, and Suranga Nanayakkara. 2018. Hand Range Interface: Information Always at Hand with a Body-Centric Mid-Air Input Surface. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Barcelona, Spain) *(MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, Article 5, 12 pages. https://doi.org/10.1145/3229434.3229449

[49] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. 2018. VirtualGrasp: Leveraging Experience of Interacting with Physical Objects to Facilitate Digital Object Retrieval. 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, 1–13. https://doi.org/10.1145/3173574.3173652

[50] Naoki Yanagihara, Buntarou Shizuki, and Shin Takahashi. 2019. A Comparative Study of Planar Surface and Spherical Surface for 3D Pointing Using Direct Touch. In *25th ACM Symposium on Virtual Reality Software and Technology* (Parramatta, NSW, Australia) *(VRST '19)*. Association for Computing Machinery, New York, NY, USA, Article 42, 2 pages. https://doi.org/10.1145/3359996.3364814