Pervasive freeform user interfaces

Aziz Niyazov

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Aziz NIYAZOV

Le 25 octobre 2023

Interfaces utilisateur ubiquitaires à formes libres

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Marcos SERRANO et Loïc BARTHE

Jury

M. Wolfgang STUERZLINGER, Rapporteur
M. Gilles BAILLY, Rapporteur
Mme MAUD MARCHAL, Examinatrice
M. Marcos SERRANO, Directeur de thèse
M. Loic BARTHE, Co-directeur de thèse

AZIZ NIYAZOV

PERFIN:

PERVASIVE FREEFORM INTERFACES

Acknowledgments

I want to express my heartfelt gratitude to all the amazing people who have helped me to earn my doctorate. This thesis would not have been possible without your assistance.

A special thank you goes to my primary supervisor from the Elipse team, Marcos Serrano. His guidance, support, and expertise have been instrumental in shaping my research. I am truly grateful for his patience, feedback, and valuable insights. I also want to acknowledge the contributions of my co-supervisor from the Storm team, Loic Barthe and Nicolas Mellado. Their input and support have greatly strengthened my work.

Furthermore, I am extremely grateful for collaborating with my esteemed co-authors from the DVIA lab at Monash University in Australia, Tim Dwyer, Barrett Ens and Kadek Ananta Satriadi. Their involvement and support have brought an international perspective to my research, fostering a broader understanding and strengthening its overall impact.

To my colleagues from IRIT, I want to express my sincere gratitude for stimulating discussions and the positive research environment we have shared. Your encouragement and collaborative spirit have been invaluable throughout this journey. Special thanks to my colleague Elen for her invaluable assistance and constant support during my research.

In addition, I want to convey my deepest appreciation to my family for their unconditional love, understanding, and constant support. Their belief in me has been a constant source of motivation and strength. I would also like to express my gratitude to my close friends, who have been a constant source of support and encouragement throughout my journey. They have continuously reminded me of how much progress I have made in life. I am thankful for the incredible moments we have shared, the endless laughter, and the deep conversations. Lastly, I want to take a moment to thank myself by citing Snoop Dogg:

"Last but not least, I wanna thank me
I wanna thank me for believing in me
I wanna thank me for doing all this hard work
I wanna thank me for, for never quitting"

Abstract

In a world of data, pervasively-displayed interfaces (also referred to as "displays everywhere") will be essential to provide access to data anytime and anywhere. The vision of displays everywhere could benefit from data visualisations that could adapt to the freeform surfaces of the surroundings, leading to the concept of pervasive freeform interfaces. Such freeform interfaces could be of interest in multiple domains, such as smart buildings, university classrooms, museums, showrooms, shop fronts and maintenance support.

This PhD focuses on developing the interaction paradigm of pervasive freeform interfaces, i.e. interfaces presented in non-rectangular display areas of the user's environment. We are mostly interested in two forms of pervasive freeform interfaces: projection-based and immersive augmented reality. Due to their characteristics, these interfaces bring new challenges to content organisation and interaction. These challenges encompass several aspects, such as interface adaptation to a freeform display area, dynamic change of display area due to the presence of physical objects, addressing the limitation of user control over layout optimisation and enabling spatial interaction to control interface layout in immersive environments.

To tackle the challenges of interface adaptation to a freeform display area and its dynamic change, we propose a novel solution based on two core contributions: the decomposition of the interface into deformable graphical units, called Dynamic Decals and the control of their position and behaviour by a constraint-based approach. Our approach dynamically deforms the interface when needed while minimising the impact on its visibility and layout properties. To do so, we extend previous work on implicit deformations to propose and experimentally validate functions defining different decal shapes and new deformers modelling decal deformations when they collide. Then, we interactively optimise the decal placements according to the interface geometry and their interrelations. This optimisation is often based on modelling the intended content placement as constraints, defined as cost functions. Applying a cost minimisation algorithm leads

to a desirable placement. Relations are modelled as constraints, and the interface evolution results from an easy and efficient to solve minimisation problem.

To address the limitations of user control over layout optimisation and controlling it in immersive environments, we explore the concept of user-driven constraints for augmented reality layout optimisation. Our approach lets users 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 study results, 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 simultaneously to arrange a set of virtual contents.

Abstract (français)

Dans un monde de données, les interfaces pervasives ("displays everywhere") seront essentielles pour permettre l'accès aux données à tout moment et en tout lieu. La vision des displays everywhere pourrait permettre de visualiser des données qui s'adapteraient aux surfaces de forme libre de l'environnement, ce qui a conduit au concept d'interfaces de forme libre pervasive. Ces interfaces présentent un intérêt dans de nombreux domaines, tels que les bâtiments intelligents, les salles de classe, les musées, les salles d'exposition, les devantures de magasins et l'assistance à la maintenance.

Ce doctorat se concentre sur le développement du paradigme d'interaction des interfaces pervasives de forme libre, présentées dans des zones d'affichage non rectangulaires de l'environnement de l'utilisateur. Nous nous intéressons principalement à deux formes d'interfaces pervasives de forme libre : les interfaces basées sur la projection et la réalité augmentée immersive. En raison de leurs caractéristiques, ces interfaces posent de nouveaux défis en matière d'organisation du contenu et d'interaction. Ces défis englobent plusieurs aspects, tels que l'adaptation de l'interface à une zone d'affichage de forme libre, le changement dynamique de la zone d'affichage en raison de la présence d'objets physiques, la prise en compte de la limitation du contrôle de l'utilisateur sur l'optimisation de la mise en page et l'activation de l'interaction spatiale pour contrôler la mise en page de l'interface dans les environnements immersifs.

Pour relever les défis de l'adaptation de l'interface à une zone d'affichage de forme libre et à son changement dynamique, nous proposons une nouvelle solution basée sur deux contributions essentielles : la décomposition de l'interface en unités graphiques déformables, appelées "Dynamic Decals", et le contrôle de leur position et de leur comportement par une approche basée sur les contraintes. Notre approche déforme dynamiquement l'interface lorsque cela est nécessaire tout en minimisant l'impact sur sa visibilité et ses propriétés de mise en page. Pour ce faire, nous étendons les travaux antérieurs sur les déformations implicites en proposant et en validant expérimentalement des fonctions définissant différentes formes

de decals et de nouveaux déformateurs modélisant les déformations des decals lorsqu'ils entrent en collision. Ensuite, nous optimisons de manière interactive le placement des decals en fonction de la géométrie de l'interface et de leurs interrelations. Cette optimisation est souvent basée sur la modélisation de l'emplacement prévu du contenu sous forme de contraintes, définies comme des fonctions de coût. L'application d'un algorithme de minimisation des coûts conduit à un placement souhaitable. Les relations sont modélisées comme des contraintes et l'évolution de l'interface résulte d'un problème de minimisation facile et efficace à résoudre.

Pour répondre aux limites du contrôle de l'utilisateur sur l'optimisation de l'agencement et le contrôler dans des environnements immersifs, nous explorons le concept de contraintes pilotées par l'utilisateur pour l'optimisation de l'agencement en réalité augmentée. Notre approche permet aux utilisateurs de définir et d'établir leurs propres contraintes directement dans l'environnement réel. Nous présentons d'abord un espace de conception composé de trois dimensions : les contraintes, les régions d'intérêt et les paramètres de contrainte. Ensuite, nous explorons les gestes d'entrée qui peuvent être employés pour définir les contraintes de notre espace de conception en fonction de l'utilisateur par le biais d'une étude d'élicitation. En utilisant les résultats de l'étude, nous proposons une conception et une mise en œuvre holistiques du système démontrant nos contraintes pilotées par l'utilisateur, que nous évaluons dans une étude utilisateur finale où les participants devaient créer plusieurs contraintes simultanément pour arranger un ensemble de contenus virtuels.

List of Figures

Contents

1

Introduction

Human-Computer Interaction (HCI) is a multidisciplinary field focusing on how humans interact with computers. This PhD work lies within the HCI research field and applies Computer Graphics (CG) methods for ubiquitous interfaces, i.e. the interfaces that can be displayed anytime and everywhere. With the guidance and expertise of my supervisors in these research areas, this thesis work was completed through a collaborative effort between the Elipse and Storm research groups at IRIT lab, University of Toulouse 3. Some of my work was conducted in collaboration with the DVIA lab at Monash University in Melbourne, Australia.

1.1 Context and General Challenges

Ubiquitous computing corresponds to the notion of accessing information anywhere and anytime. Ubiquitous computing or Ubicomp is a term originated by Mark Weiser. In his paper "The computer for the 21st century" [Weiser, 1999], he envisioned a future where everyday objects are augmented with computational capabilities allowing for seamless interaction via natural gestures, senses and speech.

1.1.1 *Pre-ubiquity Era*

The history of ubicomp started with the concept of "smart desktops" introduced by Vannevar Bush in 1945 [Bush, 1996]. According to his vision, smart desktops should allow individuals to instantly and intuitively access and process data. However, interactive input/output devices were not yet available in the early years of computing; thus, computers frequently used program switches and lights to communicate with their users (Figure 1.1). This was adequate for batch systems that ran one program at a time, often with the programmer acting as the operator. As lights and switches could be tested and set with just one machine command, this offered the benefit of reduced operating costs. However, later a text input window called the system console was added to enable communication between the operator and the system. From the 1960s onwards, user interaction with computers was mainly performed with the help of text-based Command Line Interfaces (CLI), with the primary input device being a keyboard. Learning the commands and how they work was a barrier to some people and, generally, not very intuitive for novice users [Chen and Zhang, 2007]. Nevertheless, some CLIs are still commonly used today, mainly by developers and system administrators, where speed and precision are crucial factors.

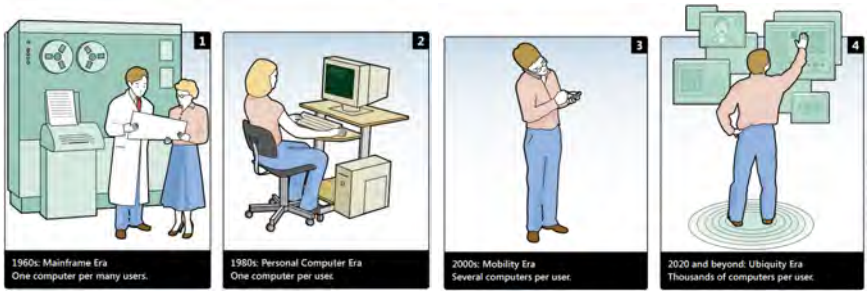


Figure 1.1: The evolution towards ubiquitous computing: 1) 1960s: Mainframe Era - one computer per many users; 2) 1980s: Personal Computer Era - one computer per user; 3) 2000s: Mobility Era - several computers per user; 4)2020 and beyond: Ubiquity Era - thousands of computers per user [Harper et al., 2008]

With further technological advancements, new WIMP GUIs (Graphical User Interfaces based on Windows, Icons, Menus and Pointers) changed how people perceive information and interact with desktop computers [Norman and Draper, 1988]. For example, WIMP GUIs allow more input capabilities due to various interaction devices, primarily a mouse, trackpad, touchscreen or graphics tablet. Compared to CLIs, WIMP GUIs were considered more user-friendly interfaces. Shneiderman et al. [2016] consider the term "user-friendly" as vague and misleading, suggesting establishing measurable criteria, such as time to learn, speed of task performance or subjective user satisfaction. Nonetheless, van Dam [1997] defines user-friendliness as easy to use. Moreover, van Dam [1997] describes the other advantages of WIMP GUIs, such as being easy to learn and the ability to easily transfer knowledge between applications due to the consistency of the interface. Being a more intuitive and easy-to-use interface [Scragg, 1982], WIMP GUIs make it easier for people to use computers and digital devices for everyday tasks [Chin et al., 1988]. Another advantage of WIMP GUIs is they are more efficient for non-technical users, as they provide high-level interface representation, such as icons or menus [Chen and

Zhang, 2007]. Yet, developing well-designed GUIs requires proper information appearance [Shneiderman et al., 2016] as a large amount of data is presented visually. Not to mention, well-designed interfaces can improve productivity, reduce errors, and enhance user satisfaction [Chin et al., 1988, Camargo et al., 2018], as well as provide more simple and quick access to information. This becomes especially important with the consistent use of multiple interfaces on a daily basis.

1.1.2 *Ubiquity Era*

Compared to desktop environments, ubiquitous computing refers to using any device in any location. Ubiquitous computing explores multimodal and multiuser interactions not limited to laptop computers, mobile phones, tablets or wearable devices. For example, Brudy et al. [2020] introduced SurfaceFleet to explore cross-device interactions not bounded to device, application, user, and time. Moreover, Houben et al. [2014] explored an activity-centric space of multifunctional devices to provide ubiquitous access to information (Figure 1.2). Furthermore, ubiquitous computing supports the creation of post-WIMP interfaces [Jetter, 2013] that enhance user experiences and performance [Poor et al., 2016]. The term "post-WIMP" was introduced by Andries van Dam and refers to alternative ways of information presentation and user interaction: *"A post-WIMP interface to me is one containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons*" - van Dam [1997]. Instead, post-WIMP UIs rely on natural language communication, such as gestures and speech recognition and focus on design principles to reflect ubicomp objectives, such as reducing cognitive effort during the interaction [Tinnell, 2016].

Figure 1.2: ActivitySpace: with a master device (laptop), the information can be moved between devices or stored in the shared space. Moreover, the system configuration may be preserved while allowing for device mobility by pinning devices in the environment. [Houben et al., 2014]

Moreover, the ubiquitous computing paradigm allows interfaces to be seamlessly integrated into everyday objects and environments, such as public spaces, homes and offices. For example, in smart homes, TVs with remote control, picture frames, mobile phones, touch screens, stereos and PCs might be used in a complementary way [Blumendorf et al., 2010]. Following the seminal ubiquitous vision, Raskar et al. [1998] introduced the concept of "the office of the future" (Figure 1.4), in which they proposed to project interfaces everywhere within a workspace. According to such a vision, anything can be a display surface within the real-world environment. This concept aimed to create an adaptive and personalised environment for individual users.

The terms ubiquitous and pervasive interfaces are often used interchangeably. This concept has recently gained popularity with an increased

need for accessing data due to dynamic interactive content and the consistent interaction vocabulary across multiple devices, for instance, using touch gestures in public displays or car interfaces. The notion of pervasive interfaces or displays everywhere [Pinhanez, 2001] covers many use cases, such as smart buildings, university auditoriums, museums, showrooms, and shop fronts. For example, to facilitate presentation meetings, brainstorming or education and enhance the interactive experience, Cotting et al. [2006], Jones et al. [2013] introduced projection-based environment-aware interfaces. Moreover, projections could be used for movable interactive devices [Lee et al., 2005, Ramakers et al., 2014] to design and explore novel form factors [Brockmeyer et al., 2013]. In addition, pervasive interfaces could be used in the context of maintenance to assist technicians [Saidi et al., 2022, Platonov et al., 2006], including remote incident playback and simulations for future buildings [Prouzeau et al., 2020]. The vision of pervasive interfaces may also impact millions of users as it can present various data visualisations and information in many domains, such as education, entertainment, healthcare or forensics [Pooryousef et al., 2023].

1.1.3 Pervasive Freeform Interfaces

Pervasive interfaces are not limited to traditional rectangular display areas. They use unconventional freeform shapes, such as circular or triangular and have been explored in various fields. Such freeform interfaces are commonly used in commercial circular smartwatches and allow novel input methods, such as bezel interaction [Neshati et al., 2021]. Another commercial example of a freeform interface is an iPhone 14. In these smartphones, Apple introduced a Dynamic Island - an area on the screen that can change in shape and size to display alerts and notifications. In addition, freeform interfaces are used in car dashboards as they have functional advantages to fit a particular environment. Other scenarios may involve interface integration in cooktops, mirrors, and bike handles [Serrano et al., 2016] and present information in circular and triangular road signs (Figure 1.3). These freeform interfaces can also be projected anywhere in the environment, allowing great versatility in choosing the optimal location for their projection. In pervasive contexts, the interfaces are subject to dynamic changes with regard to the surroundings. For instance, when a user adds a physical object, the interface adapts to the occlusion. Hence, environment-aware adaptive interfaces provide efficient space coverage and visibility of the virtual content [Riemann et al., 2018]. Besides, non-rectangular interfaces provide more creative opportunities for designers and an engaging user experience for users [Lu et al., 2020, Basballe and Halskov, 2010]. Non-rectangular interfaces enable new types of applications that were not possible before, such as interactive installations and museum exhibitions [Lu et al., 2020, Lee et al., 2019].

Pervasive freeform interfaces can be used in more immersive environments and provide an infinite canvas for virtual content (Figure 1.4). These interfaces can be fully immersed within a virtual reality or immersive augmented reality (AR) using see-through head-mounted displays (HMDs)

Figure 1.3: Some application scenarios for freeform pervasive interfaces: using a pocket mirror for private information (left), cooktop for displaying recipes (middle), warning road signs to display information for the public (right) [Serrano et al., 2016]

with CG overlays [Milgram et al., 1994]. The interaction in such immersive environments differs from traditional non-immersive 2D interfaces. To spatially organise virtual content requires different input modalities. For example, AR technologies using HMDs allow hand gestures to interact with virtual content. Besides manual interaction, the virtual content placement can be optimised and organised automatically. Such automatic optimisations of the virtual content no longer require any user input and can be based on different constraints such as semantic association [Cheng et al., 2021], user perspective [Fender et al., 2018], geometry of the environment [Ens et al., 2015] or content persistence over time [Fender et al., 2017].

Figure 1.4: Example of Ubicomp. A concept of The Office of the Future, where interfaces are projected on the horizontal and vertical surfaces [Raskar et al., 1998]

1.1.4 *Emerging Challenges*

This PhD work is part of the PERFIN (PERvasive Freeform INterfaces) ANR JCJC project, which focuses on creating an interaction paradigm for user interfaces not presented in rectangular display shapes. In this work, we are mostly interested in two forms of pervasive freeform interfaces: projected and immersive augmented reality interfaces (Figure 1.5). As illustrated earlier, these interfaces are characterized by their **freeform display area, dynamic content**, **spatial interaction** and **complex content layout**, which has led to the use of **layout optimization-based approaches**. As such, these interfaces bring new challenges in content organisation and interaction, as described in Section 1.2. To address these challenges, there is a need to reconsider the core HCI principles gathered over the years, presenting and interacting with content on rectangular interfaces. Many of these principles need to be redesigned to support non-rectangular interfaces. To reach the PERFIN objectives, this PhD addresses the general challenge of how to arrange the virtual content on pervasive freeform interfaces.

Figure 1.5: An interface projected on a plate (left) and augmented reality interface controlled by hands (right)-source.

1.2 *Challenges and Research Questions*

To tackle the main challenge of organizing virtual content in pervasive freeform interfaces, we derive it into four concrete challenges, where each challenge unfolds incrementally as the previous one is addressed.

Challenge 1 - Interface adaptation to a freeform display area. The deployment of freeform interfaces is challenged by the non-rectangularity of the display surface (e.g. a circular table), as well as the presence of objects (e.g. a cup or book on a table) that can occlude the content depending on the viewing perspective. Adapting current interfaces to fit better and take profit from the available space is complex, as traditional GUIs windows and widgets are primarily rectangular and designed for rectangular screens. This challenge leads to the first research question **(RQ1): How to adapt the interface to freeform display areas (outer shape) and the presence of physical objects that can occlude the interface (inner holes)?**

Challenge 2 - Dynamic change of the display area. In such pervasive contexts, the interface should be able to adapt dynamically to the available display space. This poses an additional problem for the outer shape or inner holes, for instance, when the user moves a cup over the interface.

These challenges lead to the following research question **(RQ2): How to dynamically adapt the interface layout to a change in outer shape or inner holes?**

Challenge 3 - Lack of user control over optimisation. Allowing users to control the content organisation is critical for efficient access to the data. However, manually arranging virtual content in the surrounding real world is complex and tedious. In contrast, automatically optimising the content placement removes the need for any user input but lacks 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 [Liu et al., 2020b]. Hence, to ensure user agency, the need for "human in the loop" [Williams et al., 2016, Roy et al., 2019] raises the following research question **(RQ3): How to bring interactivity into layout optimisation systems?**

Challenge 4 - Spatial interaction to control the interface layout. Immersive environments require different modalities for spatial interaction. For instance, AR head-mounted displays enable users to organise virtual content using hand gestures. However, such technologies rely on the 'legacy' of GUI components, such as contextual menus and UI widgets. Hence, there is a need for holistic, fluid [Elmqvist et al., 2011] and intuitive [Jacob et al., 2008] interaction that allows any user to guide the layout optimisation in the real world quickly. This leads to a new research question **(RQ4): How to provide a set of spatial gestures to control layout optimisation while not depending on the 'legacy' of GUI?**

1.3 Contribution

Our contributions incrementally address the previous four research questions in pairs in two complementary works. Our first work on 2D projected interfaces, Dynamic Decals (Figure 1.6), addresses the research questions RQ1 and RQ2. It serves as the foundational step in our research, simplifying the problem and laying the groundwork for further exploration. Our second work in augmented reality, User-driven Constraints (Figure 1.7), answers the research questions RQ3 and RQ4 and focuses on the immersive aspects, gradually progressing from simpler to more complex concepts.

To tackle research question 1 (*RQ1 - How to adapt the interface to freeform display areas (outer shape) and the presence of physical objects that can occlude the interface (inner holes)?*), we contribute to decomposing the GUI into smaller graphical elements called decals. The decals can be automatically deformed when colliding with each other, with physical objects or with the boundaries of the display window. This shape deformation allows the interface content to become freeform and maximises space occupation. The deformation is done using field function deformers. We introduce four new deformers for modelling UI objects when they collide based on two inside behaviours (squash or overlap) and two boundary behaviours (union or blending). We also implement three different initial shapes for the decals:

Figure 1.6: Dynamic Decals: Pervasive Freeform Interfaces using Constrained Defofmable Graphical Elements

circular, rectangular and rectangular with rounded corners. Moreover, we used three types of decal content (icons, images and text); however, the general approach is valid for any other content.

To address research question 2 (*RQ2 - How to dynamically adapt the interface layout due to the presence of physical objects?*), we propose a constraint-based approach. We introduce a novel set of constraints to ensure that the overall layout of decals complies with adequate layout properties. Constraints are defined as cost functions measuring specific interface properties, e.g., decals must stay in the display area and not overlap with physical objects, and decals should stay at a given minimum distance. When the user interacts with the UI or when the display area is changed, we optimise the placement of the decals by minimizing the cost introduced by this change, as measured by the constraints.

To explore research question 3 (*RQ3 - How to bring interactivity into layout optimisation systems?*), we propose a design space for user-driven constraints, i.e. constraints that can be defined or parameterized by the user, to refine augmented reality layout optimisation interactively. Our design space considers the following factors: the user-driven constraint, the constraint application region, and the constraint parameters.

To investigate research question 4 (*RQ4 - How to provide a set of spatial gestures to control layout optimisation while not depending on the 'legacy' of GUI?*), we conducted a gesture elicitation study to find the gestures that would allow the user to define, all at once, the constraint, its parameters,

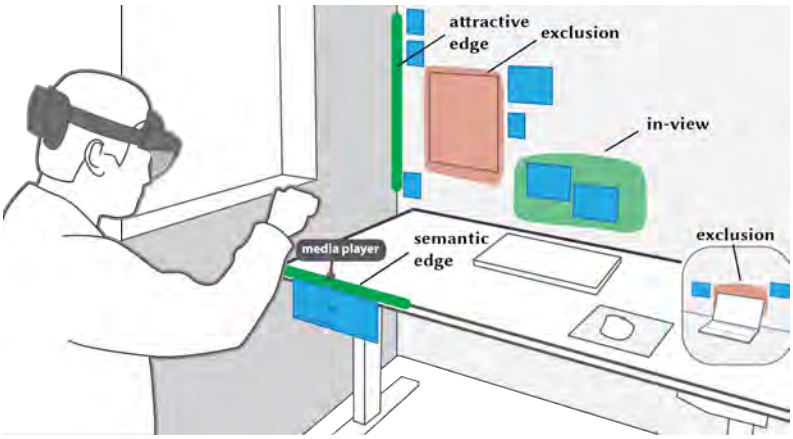


Figure 1.7: User-driven Constraints for layout optimisation in Augmented Reality

and its applied spatial region. Our approach is to move away from the inherited GUI interfaces that tend to populate augmented reality platforms, as they break the interaction flow and distract the user's attention from the surrounding real world [Jetter et al., 2014]. Our approach is also accessible to users with limited expertise in constraint optimisation.

1.4 Methodology and Approach

We adopt the iterative research process which is widely employed in HCI to design, evaluate and refine interactive systems. While we extensively apply user-centred evaluation, we also employ other types of validation approaches more common in Computer Graphics, such as controlled automatic evaluation. The iterative approach emphasises continuous feedback and improvement by iterating through multiple design iterations based on user insights and preferences. It also allows for the identification and resolution of usability issues, ensuring that the final system meets the needs and expectations of the users.

Our first work, Dynamic Decals (Figure 1.6), lies at the intersection of HCI and Computer Graphics and answers the research questions RQ1 and RQ2. To design, develop and evaluate our Dynamic Decals, we adopted the following approach:

1. We defined a novel type of deformable UI widget called Dynamic Decals.
2. We designed and implemented the graphical behaviour of Dynamic Decals. We then validated the implementation through a user study asking participants to rate the deformations that resulted from applying the four deformers to the aforementioned decals' shape and content according to different visibility and aesthetics metrics.
3. We designed and implemented a constraint-based approach to control the placement of decals. To validate these constraints through a controlled automatic evaluation, we implemented various interfaces and

emulated either a physical object occluding its content or a change in the overall display shape.

4. We validated the Dynamic Decals approach through a user study comparing our approach with two baselines, when a physical object occludes the content and when the interface fits a freeform display area.

Our second work on User-driven Constraints (Figure 1.7) uncovers the answers to the research questions RQ3 and RQ4. In summary, our development approach is the following:

1. We created a design space of interactive layout optimisation for augmented reality environments. Our design space considers the following factors: the user-driven constraint, the constraint application region, and the constraint parameters.

2. We conducted a gesture elicitation study to explore this design space, focusing on gestures users would perform for such interactive optimisation. The participants proposed gestures for a set of combinations of constraints, parameters and regions while wearing an HMD.

3. 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.

4. We validated our approach through a controlled summative study, where participants had to create several constraints at the same time to arrange a set of virtual contents.

1.5 Structure of Manuscript

This manuscript comprises five chapters, including this Introduction Chapter 1. In Chapter 2, we provide a literature review on the layout organisation and adaptation in static and dynamic interfaces, layout optimisation techniques and interaction in immersive environments. In Chapter 3, we present Dynamic Decals, a novel approach to overcoming the challenges of pervasive interfaces caused by freeform display area and object occlusion. In Chapter 4, we present User-driven Constraints by addressing the challenges of letting users define constraints to optimise virtual content placement in immersive AR environments. Finally, in Chapter 5, we conclude this manuscript by summarising presented contributions and discussing the perspectives of future work.

2

Background and Related Works

Deploying pervasive freeform interfaces represents a significant shift from the conventional practice of displaying and interacting with content on rectangular screens and raises fundamental design questions. In the past few decades, researchers in HCI have developed multiple techniques to address the challenges of virtual content arrangement in such freeform interfaces. The solutions cover a wide range of scopes, from a content organisation in interfaces having any 2D shape to more advanced information presentation techniques in immersive environments.

In this chapter, we first provide an overview of the background of freeform interfaces (Section 2.1). These freeform interfaces raise questions about content presentation due to their non-rectangularity. To address these challenges, we revised techniques of layout organisation in static interfaces (Section 2.2). Understanding content composition in traditional rectangular interfaces helped to organise the layout in freeform interfaces. The layout adaptation and organisation are essential to comprehend the relations between virtual elements, especially in a dynamic context. For instance, the presence of physical objects or the perspective from which the content is viewed requires content repositioning. Previous works presented dynamic layout organisation (Section 2.3) from flat interfaces to

immersive environments, such as AR. In addition, other approaches were proposed to organise virtual content dynamically using layout optimisation techniques (Section 2.4). Such layout optimisations are beneficial but lack user control in refining constraints. Human involvement is critical as it brings interactivity in layout optimisation systems and ensures user agency, especially in more immersive domains such as AR environments (Section 2.5).

2.1 *Background on Freeform Interfaces*

The term "Freeform interfaces" was proposed by Serrano et al. [2016] and referred to interfaces with non-rectangular shapes. Freeform interfaces take their roots in Organic User Interfaces (OUIs) and Shape-changing Interfaces (SCIs). OUIs were introduced by Vertegaal and Poupyrev [2008] and refer to adopting natural forms to create a more suitable environment for humans. Holman and Vertegaal [2008] defined them by three main principles: "input equals output" or Flexible UIs, "function equals form" or Shaped UIs, and "form follows flow" or Actuated UIs (Figure 2.1). The first design principle, "input equals output", means that displays should follow the shape of the object, and the input method should correspond to such display shape. This behaviour can be found in current smartphone technologies using multitouch interaction (Figure2.1 - left). The second principle, "function equals form", resonates with Gibson's affordance [Gibson, 1979] and states that the form of the object should determine its function. For instance, the illuminated face of an icosahedron may represent a selected menu item (Figure2.1 - centre). The third principle, "form follows flow", expresses interface adaptation to follow user activities and better fit a particular context fluidly. For example, a moldable mouse with jelly anatomy (Figure 2.1 - right).

The latter principle coincides with the notion of Shape-changing Interfaces, which has its roots in Sutherland's Ultimate Display, where he describes a computer that can "control the existence of matter" [Sutherland et al., 1965]. Following this vision, Alexander et al. [2018] described the fundamental nature of Shape-changing Interfaces. The authors defined SCIs as self-and/or user-actuated interfaces that convey information, meaning, or effect. Such interfaces use a physical change of shape or materiality as input and/or output whilst being interactive and computationally controlled. While shape-changing and freeform interfaces may naturally integrate with the human environment, they raise fundamental questions about how well to organise rectangular content with regard to their shape features.

Summary of Background on Freeform Interfaces

The term "Freeform interfaces" refers to interfaces with non-rectangular shapes. Such shape features allow interfaces to naturally integrate with the human environment but raise fundamental challenges regarding content organisation.

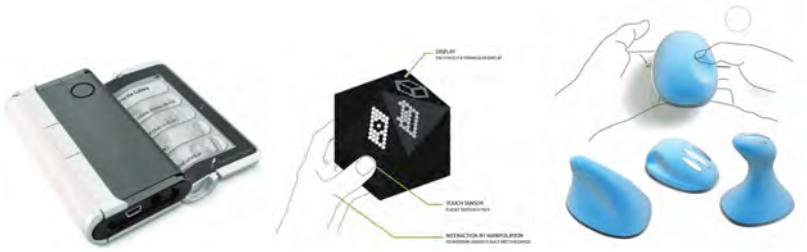


Figure 2.1: Readius: First foldable cell phone with e-paper display (left) [Vertegaal and Poupyrev, 2008], D20: Multi-faceted handheld display device (middle) [Poupyrev et al., 2006], Moldable mouse (right) [Vertegaal and Poupyrev, 2008, Holman and Vertegaal, 2008]

2.2 *Layout Organisation in Static Interfaces*

Layout organisation refers to the arrangement and organisation of elements of an interface, such as text, images, and other visual elements. A well-organised layout can enhance the user's experience by making it easier to perceive and interact with the content [Shneiderman et al., 2016, Ware, 2004]. However, to design and organise content for freeform interfaces, it is first essential to understand how content is organised and perceived in rectangular interfaces. Thus, we initially present layout guidelines for traditional rectangular interfaces and subsequently demonstrate how they are applied to freeform interfaces.

2.2.1 *Rectangular Layouts*

Layout organisation was first studied for standard rectangular displays, such as desktop or mobile displays, to define one GUI that can adapt to multiple target platforms [Stuerzlinger et al., 2006, Meskens et al., 2008]. To keep the relations between content and perceive it the same way while migrating from one device to another requires preserving core design principles, many of which heavily rely on Gestalt Psychology. Gestalt Laws suggest how static visual elements should be presented to achieve effective visual results. Helson [1933] extracted 114 laws of Gestalten, and the majority of them apply to visual form [Woodworth, 1942].

The Gestalt Laws are applicable to layout organisation to enhance the visual composition and aesthetics of the GUI [Dondis, 1974]. Moreover, they could be complemented by the golden ratio to divide the layout into pleasing proportions [Huntley, 1970]. In addition, the Gestalt principles of perception include the Law of Prägnanz, which according to Lidwell et al. [2010] implies to *"when people are presented with a set of ambiguous elements (elements that can be interpreted in different ways), they interpret the elements in the simplest way"*. Such gestalt principles were widely used in GUI design and layout perception [Flieder and Mödritscher, 2006, Fraher and Boyd-Brent, 2010, Koch and Oulasvirta, 2016]. For example, Chang et al. [2002] described 11 laws of Gestalt Theory that are heavily related to the Law of Prägnanz and significantly impact the design of computer screens: balance/symmetry, continuity, closure, figure-ground, focus, isomorphic correspondence, good form, proximity, similarity, simplicity and unity/harmony. Yang and Klemmer [2009] used the elements of Gestalt

Theory, such as balance/symmetry, simplicity, and proportion, in web page design to improve user satisfaction. Applying Gestalt Laws, Koch and Oulasvirta [2016] explored the perception of interactive layouts from a computational perspective by identifying the grouping of visual elements.

Additionally, Galitz [2007] proposed 10 gestalt-based aesthetic qualities to define layout properties (Figure 2.2): Balance, Symmetry, Regularity, Predictability, Sequentiality, Economy, Unity, Proportion, Simplicity, and Grouping. Such properties provide a visually pleasing layout composition and affect usability, which defines how easy an interface is to use. Inspired by these aesthetic properties, Ngo et al. [2003] developed a framework for objective automatic evaluation of 14 metrics to model interface aesthetics: balance, equilibrium, symmetry, sequence, cohesion, unity, proportion, simplicity, density, regularity, economy, homogeneity, rhythm, and order and complexity.

Previously mentioned aesthetic qualities of gestalt principles constitute visually pleasing compositions that capture attention and convey a positive message clearly and quickly. On the contrary, a lack of aesthetics can cause confusion and disorientation, making it difficult to understand the intended message. Although these metrics are useful for evaluating the layout properties, most are limited only to rectangular interfaces.

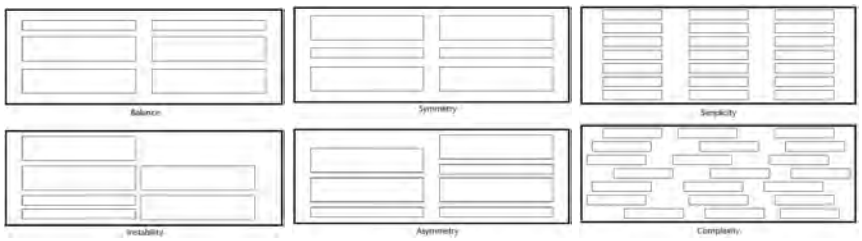


Figure 2.2: Few examples of comparing aesthetic qualities of the interface by Galitz [2007]: Balance/Instability (left); Symmetry/Asymmetry (middle); Simplicity/Complexity (right)

2.2.2 *Freeform Layouts*

Freeform interfaces belong to a particular subcategory characterised by their non-rectilinear structure. Based on Gestalt Laws, recent research on these interfaces focused on designing content that adapts to their static shape features. The guidelines for designing such freeform interfaces were explored by Serrano et al. For example, Serrano et al. [2016] explored geometrical static shapes such as circles and triangles as a first step. Then the authors investigated the effects of text mappings on arbitrary non-rectangular shapes on reading ability and perceived aesthetics. The results of this study revealed novel practices for presenting text on non-rectangular interfaces (Figure 2.3). For instance, in the case of implementing dynamic scrolling, it is advised to adjust the text size so that each line has an equal amount of text. A subsequent study [Serrano et al., 2017] focused on visual layouts and compared them in terms of perceived symmetry, clarity, and preference (Figure 2.4). The findings led to a set of design guidelines that contradict some standard criteria of UI design. For instance, designers can customize the shape of content to match the display area (such as circular

content for a circular display or triangular content for a triangular display) instead of using rectangular boxes for text or images.

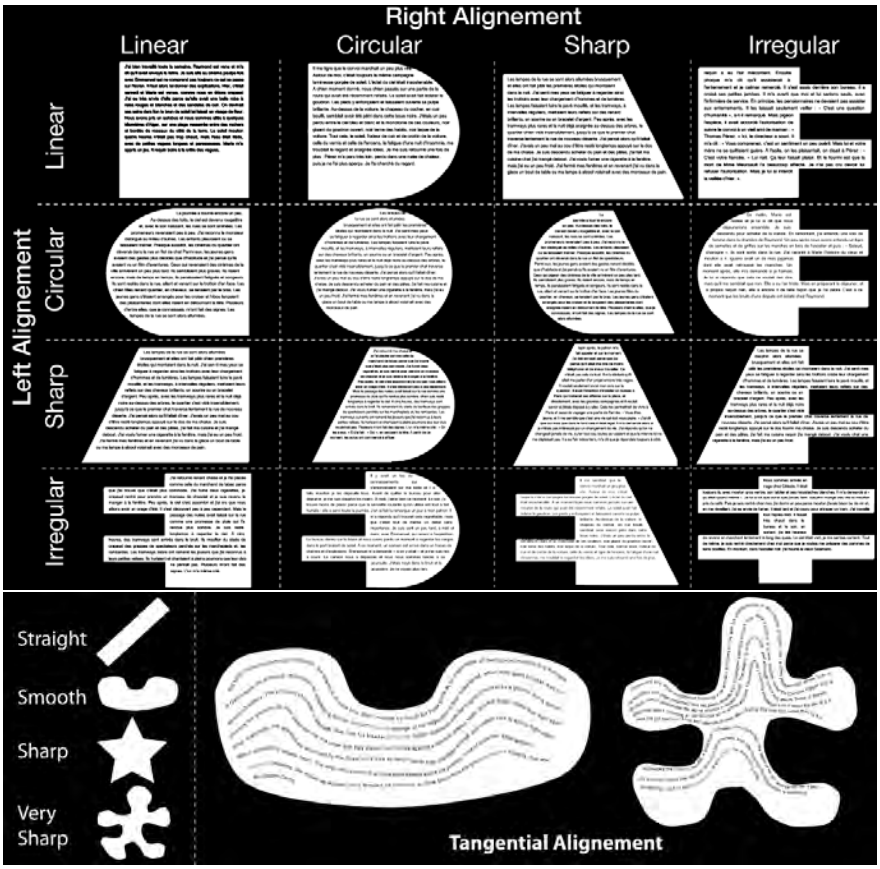


Figure 2.3: Text mapping to non-rectangular shapes with left, right and tangential alignment [Serrano et al., 2016]

Figure 2.4: Graphical design of static freeform interfaces [Serrano et al., 2017]

Extending these works, Simon et al. [2019] have examined how individuals find information on non-rectangular interfaces. They analysed eye-tracking data to determine which display areas were viewed first depending on visual layout composition. For instance, the gaze pattern follows a clockwise direction on non-rectangular interfaces (Figure 2.5). Also, their results show a decreased time to find a specific item presented on a rectangular grid, compared to a non-rectangular one, which helps designers place important content on freeform interfaces. However, this

previous research on simple geometrical shapes (circular, triangular) has focused on visual content perception rather than addressing the content adaptation challenges of freeform interfaces in dynamic contexts.

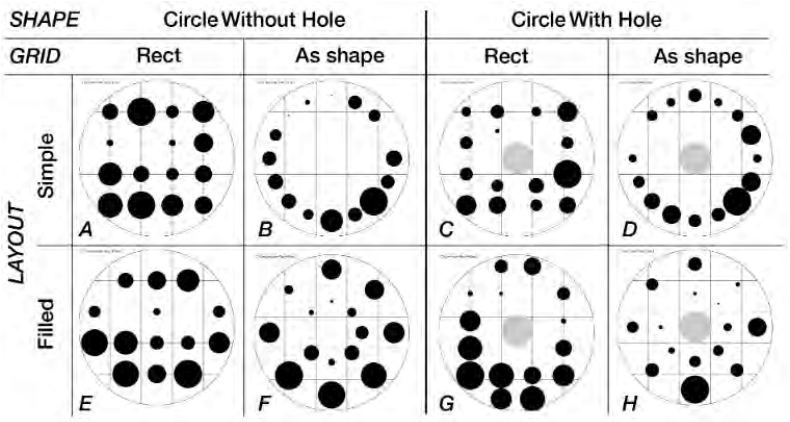


Figure 2.5: Gaze pattern on circular interfaces. (Bigger circles correspond to longer eye fixations) [Simon et al., 2019]

2.2.3 *Immersive Layouts*

Displaying information in immersive environments raises new challenges for layout organisation. In this PhD work, we focus only on the arrangement of 2D content; thus, we describe related works that mostly focus on 2D widgets.

Ens et al. [2014a] introduced a design space for 2D information spaces in augmented reality environments around a fundamental layout dimension: the reference frame (Figure 2.6). The reference frame includes the perspective and the movability of the content. Regarding perspective, the content can be arranged in either egocentric (i.e. body-based coordinates) or exocentric (i.e. world-based coordinates) perspectives. Regarding movability, the content can either move with the user or be fixed in space. Most often, the egocentric perspective is combined with movable content, such as on-body interfaces [Chen et al., 2012, Harrison et al., 2010, 2011, Gustafson et al., 2011] or hand-held palettes [Google, de Haan et al., 2002, Lindeman et al., 1999]. Such body-centric UIs follow the user as they move. Thus the organization of the contained widgets is independent of the external environment. Conversely, exocentric UIs often contain world-fixed content, as when interacting through a "peephole" that follows a spotlight metaphor [Boring et al., 2010, Cauchard et al., 2012, Fitzmaurice et al., 1993], or floating in mid-air windows [Andujar and Argelaguet, 2006, Hoang and Thomas, 2010, Chan et al., 2010]. This later case, i.e. exocentric perspective with fixed content, is relevant for information visualisation.

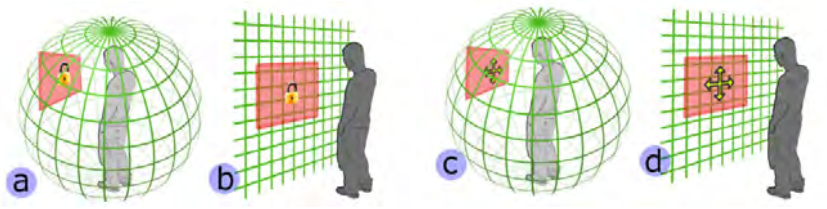


Figure 2.6: Four general Reference Frames [Ens et al., 2014a]: (a) fixed-egocentric, (b) fixed-exocentric. The (c) movable-egocentric and (d) movable-exocentric will be discussed in the next section

Summary of Layout Organisation in Static Interfaces

Layout organisation in conventional rectangular interfaces heavily relies on Gestalt Laws. These laws are applicable to enhance the visual composition and aesthetics of the interfaces. Previous works focused on layouts in static freeform interfaces with predefined shapes, leaving aside dynamic contexts (i.e. when the outer shape changes, for instance). In immersive environments, virtual content can be fixed in the environment or movable. Also, it can be presented with an egocentric or exocentric perspective. Our work focuses on the exocentric perspective and movable content, as we address the question of how to move content in accordance with user-defined constraints.

2.3 *Layout Adaptation in Dynamic Interfaces*

Content adaptation is critical in any pervasive environment, whether a tabletop, projected system or immersive environment. In this section, we first describe how virtual content can adapt to the presence of virtual and physical objects in rectangular layouts. Then, we present multiple approaches to adapting GUI elements to freeform layouts. Finally, we review previous approaches for layout adaptation in immersive environments.

2.3.1 *Rectangular Layouts*

We review previous work on content adaptation in rectangular tabletops, where the content is presented as rectangular windows, as this context is the closest to our target environments. Layout adaptation in this context looks to overcome virtual content occlusion by physical objects such as books, cups or plants [Javed et al., 2011, Tabard et al., 2013]. For instance, Javed et al. [2011] derived six techniques for managing occlusions: adding glow to physical objects' outlines (Figure 2.7), using miniature icons to represent occluded objects, combining virtual and physical objects to create a hybrid pile, providing an overview map, moving virtual objects to empty space and replicating hidden areas in unoccluded space. Their study results show that glowing performs significantly better than the other techniques. In addition, Vogel and Balakrishnan [2010] defined the concept of Occlusion-aware interfaces, i.e. interaction techniques which know what display area is occluded to counteract potential problems or use the hidden area. The usual

approach in these occlusion-aware interfaces presents the occluded content or an icon-sized miniature around the physical objects [Khalilbeigi et al., 2013, Furumi et al., 2012], which are used as anchors. For instance, ObjecTop by Khalilbeigi et al. [2013] presents an icon-sized miniature representation of the occluded content (which remains in its original location) around the physical object. However, this approach does not preserve content visibility and requires the user to perform an explicit action to access the content. SnapRail by Furumi et al. [2012] preserves content visibility by creating a circular widget rail around the physical objects where the interface elements are attached. However, this technique has the disadvantage of not preserving the original layout of the interface. Also, adopting a circular layout can make it difficult to handle large virtual elements or too many elements.

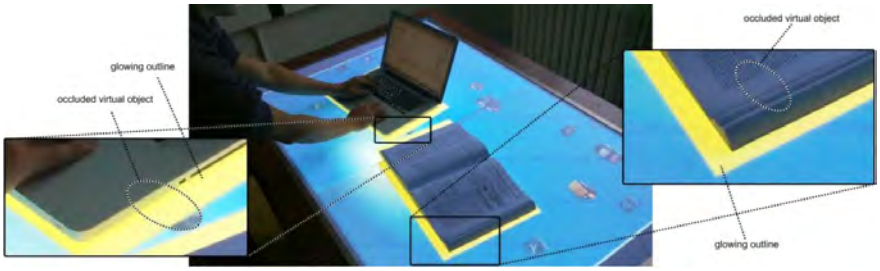


Figure 2.7: Glowing effect on virtual content occlusion [Javed et al., 2011]

Figure 2.8: Occlusion-aware interfaces: SnapToRail (left) [Furumi et al., 2012], ObjectTop (right) [Khalilbeigi et al., 2013]

2.3.2 *Freeform Layouts*

Previous works presented various approaches to adapt GUIs to non-rectangular display areas. For instance, Waldner et al. [2011] developed display-adaptive window management for irregular surfaces. Their approach semi-automatically placed the rectangular windows in the available display space. To best exploit the available freeform display space in pervasive environments, others have explored projected interfaces combined with content mapping or deformation. For instance, Ramakers et al. [2014] developed a foldable smartphone prototype by dynamically projecting the virtual content on the device. At a larger scale, the Illumiroom and RoomAlive concepts [Jones et al., 2013, 2014] are room-sized systems where the virtual content is projected onto the walls. This content distortion can also create perspective-aware interfaces [Nacenta et al., 2007].

The combination of the previous considerations, i.e. object occlusion management and content deformation, has been scarcely explored. Cotting

Figure 2.9: Illumiroom (left) and RoomAlive (right) by Jones et al. [2013, 2014]

et al. [2006] developed the Display Bubbles system, a projected interface that is deformed to adapt to the environment. However, in this work, the whole window is warped, which impacts the visibility of the content, particularly for UI elements that are close to the edges of the window (and which are more severely deformed). Controlling the deformation of the interface leads to a tradeoff between efficient space coverage and visibility/manipulability of the content. FlowPut [Riemann et al., 2018] is an environment-aware framework that projects output on and around tangible objects: the projection is optimised by environmental constraints to avoid interferences between the projection and real-world objects. The authors of FlowPut proposed two presentation techniques: to place visual elements based on environmental constraints or to project as much visual content in the surrounding tangible object as possible.

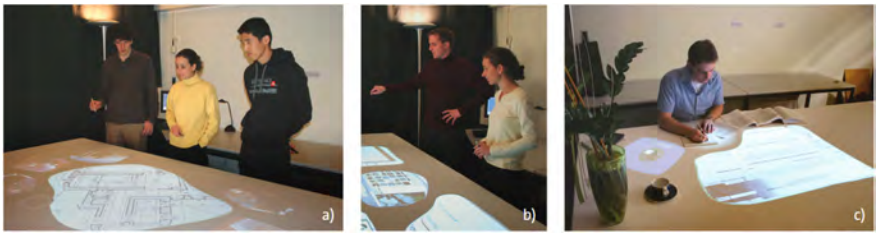


Figure 2.10: Cotting et al. [2006] introduced interactive, environment-aware display bubbles as a novel display metaphor that can be applied to a broad range of use cases and applications. These include joint development, design, modelling, and visualisation(a); group sessions for generating ideas (b); and large workspaces(b).

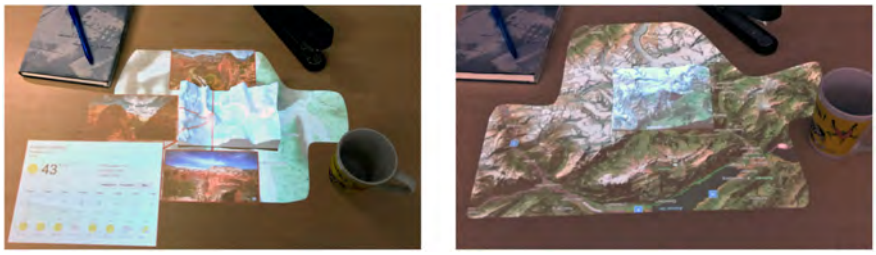


Figure 2.11: An environment-aware projection is used to augment the space around a tangible object, allowing for the placement of free-floating UI elements such as images or information widgets (left), or to expand the size of a UI element based on the surrounding environment (right) [Riemann et al., 2018].

In addition, previous research tried to define interfaces with particle systems applying some physics [Wilson et al., 2008, Langner et al., 2010, Jones et al., 2010, Agarawala and Balakrishnan, 2006]. For instance, [Wilson et al., 2008] used an advanced game physics engine combined with a touch surface to add real-world dynamics to interactive surfaces. This approach uses physical simulation to handle collisions and frictions of virtual objects. Brett R. Jones et al. [Jones et al., 2010] presented a projection-based inter-

action of particles on irregular surfaces. These approaches do not handle content overlapping nor adapt it to the layout. In BumpTop [Agarawala and Balakrishnan, 2006], physics simulation is applied to a virtual desktop to structure its objects. Although this approach handles content organisation, it does not consider overlapping conditions. Overall, physics-based methods are well-adapted to develop playful interfaces and real-world metaphors.

Figure 2.12: (a) Traditional virtual desktop layout with organised content. (b) A physical desk where items are arranged in a casual manner. (c) The BumpTop prototype uses piles to organise content and uses a physics simulation to allow for more natural and authentic interaction for a more realistic experience [Agarawala and Balakrishnan, 2006]

2.3.3 *Immersive Layouts*

As said earlier, displaying information in immersive environments raises new challenges in the layout adaptation of multiple 2D widgets. In particular, it might be tedious to reposition a large number of content over a considerable distance when the interface layout is changed.

Previous works demonstrated dynamic interface adaptation to facilitate the interaction with the virtual content. For instance, [Grubert et al., 2015] introduced a system that uses HMD to augment mobile displays (smart-watches, smartphones, tablets). Their prototype allows on-the-go interaction and provides several dynamic body-centric interface adaptation modes, such as body-aligned, device-aligned and side-by-side modes (Figure 2.13).

Another example where the interface layout moves with the user is Personal Cockpit [Ens et al., 2014b]. It allows on-the-go multitasking, such as accessing multiple applications quickly and easily and linking virtual content with related tasks. Moreover, Personal Cockpit allows switching interface layout, for example, based on viewing perspective: from a body-

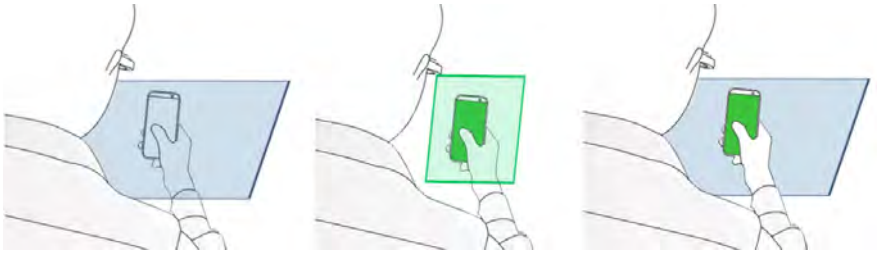


Figure 2.13: In body-aligned mode (left), gadgets are registered in a common information space with respect to a user. In device-aligned mode (middle), the display area of the smartphone is expanded. In side-by-side mode (right), devices are separated and don't need a spatial correlation [Grubert et al., 2015].

centred to a world-centred coordinate system (Figure 2.14).

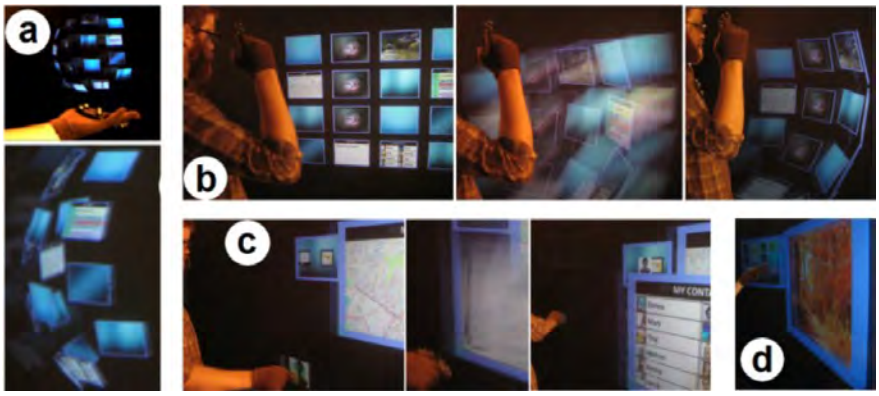


Figure 2.14: Personal Cockpit [Ens et al., 2014b] interaction scenarios include: transitioning from a body-fixed to a world-fixed layout (a), opening a new app window (b), virtual content intercommunication (c), and minimizing the interface to a small, palm-sized overview (d)

In addition, Liu et al. [2020a] explored the adaptation of 2D small-multiples visualisations in immersive environments. In their study, the authors explored the effect of layout curvature (Figure 2.16) on performance (time and accuracy). Their research revealed that Flat layouts provide a better overview of small multiples and are good for a small number of visualisations; otherwise, the walking distance increases. The Full-Circle layout requires less walking but disorients users and makes it hard to locate information. Compared to Flat or Full-Circle layouts, Half-Circle layouts provided a compromised solution and were preferred by participants. Another work by Luo et al. [2022] examined the layout strategies and their transitions in collaborative sensemaking and brainstorming activities in immersive AR environments. In particular, the authors explored the dependency of spatial exocentric layouts depending on the physical environments (Figure 2.15). Their study results showed that participants were actively using the furniture despite their personal preferences.

Previously mentioned approaches of spatial layout adaptation have been shown to be relevant for information visualisation; however, they raise the question of setting the content position by the user. Thus, in this PhD, we focus on positioning virtual 2D widgets in the user's spatial surroundings. We detail previous approaches to human interaction in Section (2.5)

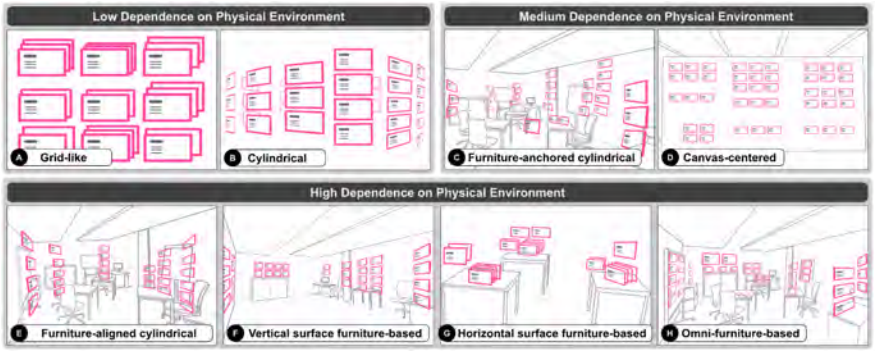


Figure 2.15: Examples of virtual content placement with regard to real-world environment [Luo et al., 2022]

Figure 2.16: Multiple possibilities for layout curvature: Flat, Half-Circle, Full-Circle (left). Small multiples presented in VR using a "shelves" metaphor (right) [Liu et al., 2020a]

Summary of Layout Adaptation in Dynamic Interfaces

Previous works explored content adaptation in a dynamic context to preserve content visibility in rectangular layouts, which however, does not maximize display space occupation. In freeform layouts, previous work proposed to deform the content to fit the freeform display area; however, this affects the visibility of the interface. Layout adaptation in immersive layouts raises the challenges of spatial content positioning, especially when there is a large number of virtual objects.

2.4 *Layout Optimisation*

Layout optimisation is essential in pervasive dynamic contexts as it allows users to arrange virtual content without the need for tedious manual interactions. However, there are no previous works on content optimisation in freeform interfaces. Thus, in this section, we first describe how virtual content is automatically organised in rectangular interfaces while preserving some constraints of layout optimisation. More specifically, we first focus on content positioning *within* the display area. Then, we expand the layout optimisation to 3-dimensional spaces. We review previous works presenting multiple constraints to arrange GUI elements in *between* multiple

display areas in immersive layouts, such as AR.

2.4.1 *Rectangular Layouts*

Standard user interfaces are usually designed manually, which limits their use in dynamic contexts. Several approaches have been proposed to optimise the interface layout and speed up the creation process. For instance, Scout [Swearngin et al., 2020] is a system helping designers to explore alternatives through high-level constraints based on design concepts such as semantic structure or emphasis. Dayama et al. [2020] propose an optimization approach to generate grid-based layouts. These methods can also integrate a modelization of the user's performance. For instance, Duan et al. [2020] presented an automatic method to optimize the layout of mobile UIs using a predictive model of task performance. Constraints have also been applied to generate web interfaces according to predefined requirements, such as image placement depending on the page width [Borning et al., 1997]. All these approaches are similar in generating static versions of the interface, either during the design process or before rendering (as for web interfaces).

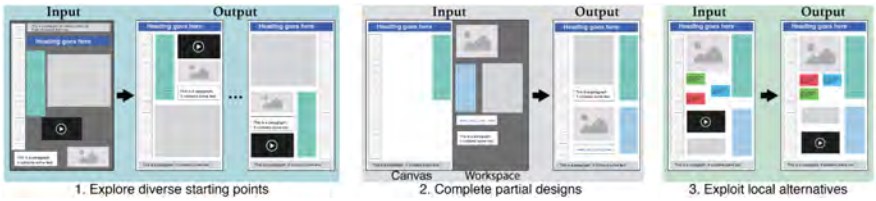


Figure 2.17: The use of integer programming allows for interactive grid layout generation, enabling designers to (1) methodically investigate various starting points, (2) discover solutions for designs that are only partially completed, and (3) explore sub-spaces to identify local alternatives and solutions [Dayama et al., 2020].

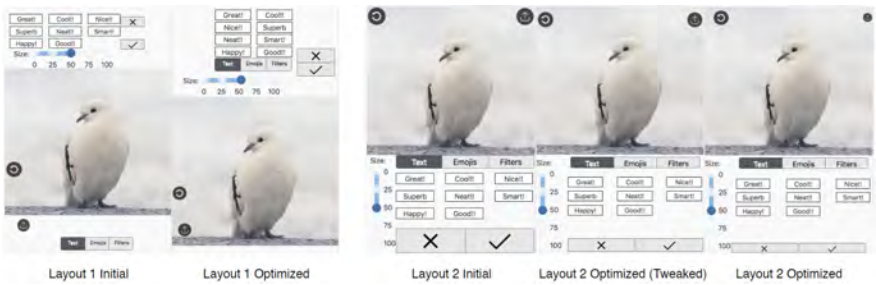


Figure 2.18: Two different layouts were optimized using different constraints. The first "Layout 1", underwent optimisation under strict constraints, and its resulting output is referred to as "Layout 1 Optimised". The second "Layout 2", was optimized under relaxed constraints, and its resulting output is referred to as "Layout 2 Optimised" [Duan et al., 2020].

Automatic methods have been proposed to adapt interfaces to user actions dynamically. In 1962 Sutherland [1964] already used atomic constraints in the seminal Sketchpad to control lines organization: vertical, horizontal, parallel, or perpendicular. Constraints were then used to assist direct manipulation techniques in interactive geometric modelling [Sistare, 1991, Gleicher, 1992a, Hudson and Smith, 1996]. For instance, Gleicher [1992a,b] used constraints to establish and preserve relationships between geometric objects positioned with direct manipulation. Most of

these approaches rely on linear constraints, which do not scale well to real-world applications. Instead, Hosobe [2001] proposed a constraint solver handling non-linear geometric constraints such as Euclidean geometric, non-overlapping, and graph layout constraints. When applied to UIs, constraints help to define the state of an interface according to user input, as in ConstraintJS [Oney et al., 2012], or to adjust layouts when the window dimensions change dynamically. For instance, Cassowary [Badros et al., 2001] is a constraint solver algorithm used in windows management in some commercial systems, such as Apple's AutoLayout.

While powerful, these constraints have limitations. They do not support flow layouts and require explicit layout specifications, for instance, when transitioning from portrait to landscape orientation. Jiang et al. [2020a] addressed these limitations and proposed the OR constraints (ORC) layout in a soft/hard linear constraint system. An ORC is essentially a disjunction of multiple constraints where only one must be true. The ORC layout system supports various layout patterns, including combinations of flow and grid layouts. It also allows layouts to adapt flexibly to different screen sizes and orientations. Furthermore, Jiang et al. [2020b] introduced an ORCSolver, specifically designed to handle complex and realistic layouts that involve linear constraints, flows, and combinations thereof. This solver is more efficient than modern constraint solvers like Branch&Bound [Markowitz and Manne, 1957], Quadratic Programming Solver [Frank and Wolfe, 1956], and Z3 [De Moura and Bjørner, 2008], providing nearly interactive performance.

However, most of these systems perform on rectangular content distributed on linear grids. Riemann et al. [2018] use environmental constraints (projection quality, proximity to the desired location, orientation, scale) to project content on and around tangible objects (Figure 2.19). However, they did not apply such constraints to reorganise traditional UI layouts. To our knowledge, no previous approach tackles the challenge of organising non-rectangular UI elements in a freeform display space.

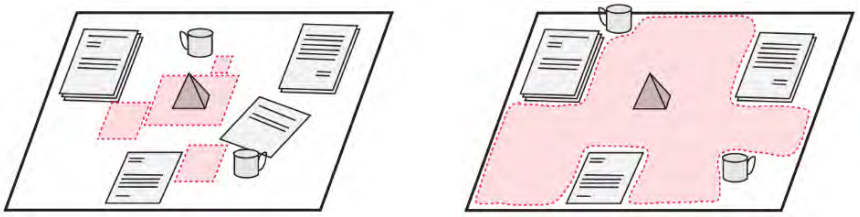


Figure 2.19: Layout for UI elements based on environmental constraints (left) and maximum space layout (right). The projection is shown in red. [Riemann et al., 2018]

2.4.2 *Immersive Layouts*

Many efforts have been made towards defining automatic layout optimisation approaches in immersive environments. The goal of these optimisation approaches is to automatically place the virtual content in desirable locations of the environment to preserve predefined constraints. This optimisation 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 [Tapal et al., 2017, Lu and Xu, 2022]. We review the constraints proposed to optimise content placement in augmented reality environments.

- **Environment geometry**: Environment geometry refers to virtual content placement with respect to the real-world environment and is one of the core requirements in most AR scenarios. Gal et al. [2014] introduced a constraint-based framework for AR applications to design environment-adaptive layouts. Their algorithm, by extracting horizontal and vertical surfaces, allows arranging virtual objects dynamically to find an optimal layout. Nuernberger et al. [2016] extracted the edges of the environment in real-time to align, rotate and scale virtual windows to real-world dynamically. The authors' approach, by visualising physical constraints, allowed them to align the virtual objects faster and position them more precisely using snapping (Figure 2.20). Moreover, snapping techniques enabled novel ways of interaction when creating AR content. For instance, door opening, chair displacement or adding objects into the environment creates new alignment opportunities.

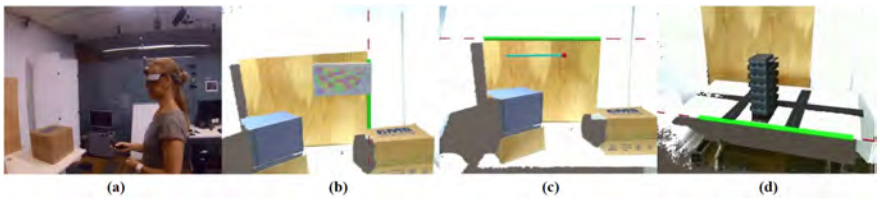


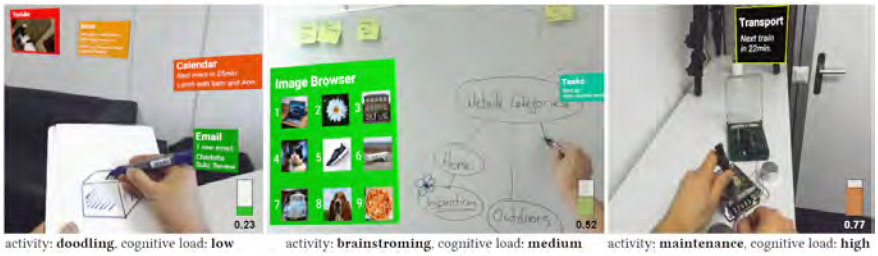
Figure 2.20: Example of environment geometry constraint: SnapToReality provides users with a simple way to align AR virtual content with real-world constraints by adjusting its position, orientation, and scale. The prototype can (a) identify and extract planar surfaces and edges from the real world, which can be used as constraints to snap virtual content into place (b, c). SnapToReality allows for AR content to be seamlessly integrated into the real world (d). [Nuernberger et al., 2016]

- **Spatial consistency**: Spatial consistency refers to the virtual content distribution in the space while maintaining its position relative to the users or other physical or virtual objects in the environment. Lu and Xu [2022] addressed the challenges of UI transition when individuals use AR interfaces on the go by designing three interfaces with different levels of automation and control. Their study shows that maintaining the spatial layout consistency can improve content memorisation across multiple environments [Lu and Xu, 2022]. In addition, Ens et al. [2015] explored the transition from egocentric to exocentric reference frames while preserving spatial consistency. To do so, the authors introduced a layout manager that maintains content position relative to a user and to other virtual objects with the help of constraints. For instance, adherence constraint minimises the angular distance of a virtual content location with respect to a user.

- **Cognitive load**: Cognitive load refers to the capacity of working memory or mental effort required to perform a particular task. Lindlbauer et al. [2019] proposed a real-time approach to automate UIs when switching tasks or environments. Such an approach allows moving from fully manual content creation or arrangement to a computationally supported one. Depending on the expected cognitive load of the ongoing task, the

Figure 2.21: Example of spatial consistency constraint: The transition of the virtual content from a body-centric layout to world-fixed coordinates (a). The spatial relationships between the elements are preserved while moving from one environment to another (b,c). [Ens et al., 2015]

system increased or decreased the number of UI elements and their level of detail (LoD). For instance, when reading a complex text that causes an increase in cognitive load, the system may reduce the number of visual augmentations and keep only text-related ones. According to study results, this approach increases task performance and the predictability of a system.



activity: **doodling**, cognitive load: **low** activity: **brainstroming**, cognitive load: **medium** activity: **maintenance**, cognitive load: **high**

Figure 2.22: Example of cognitive load constraint: the system shows more virtual content and in more detail for tasks with a low cognitive load (left), whereas less virtual content is shown with a lower level of detail for tasks with an increased cognitive load (right) [Lindlbauer et al., 2019]

- **User perspective**: User perspective refers to the viewing angle and field of view of the users, taking into consideration the occlusions of virtual objects.Fender et al. [2017] recorded and analysed users' behaviour within an environment and automatically suggested the size and position of virtual content. Depending on the users' Field of View and orientation, the system optimises content placement Fender et al. [2018]. This is especially useful in a dynamic context, where the content appearance is view-dependent. For instance, Lages and Bowman [2019] explored how to adapt the content to the ongoing location and task while walking. The authors proposed four basic behaviours of content adaptation (follow, rotation, attraction, and auto-centring) and their combinations (Figure 2.23). The combination of "follow" with "auto-centring" was the most preferred.
- **Utility and usage frequency**: Utility and usage frequency refer to the extent the virtual content is used. Cheng et al. [2021] performed optimisation considering the utility of the virtual content with respect to a specific task. Higher utility elements should be placed in more accessible locations, such as "within reach" to easily access task-relevant content or "within sight" to place important information. In addition, Lindlbauer et al. [2019] optimised the virtual content according to its utility and usage frequency to adapt how much information to show and where to place it.

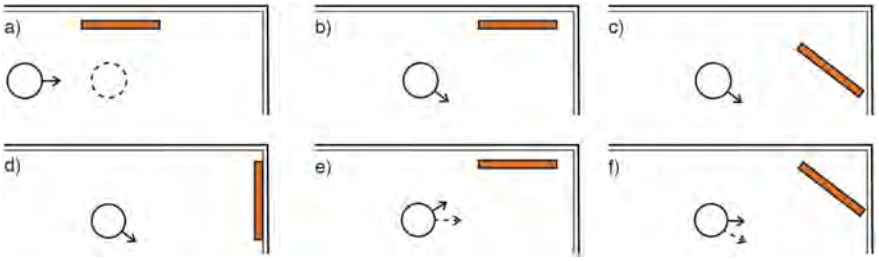


Figure 2.23: Example of user perspective constraint: the circle represents the user's location in relation to a single window (orange), while the solid arrow indicates the viewing direction, and the dashed arrow represents the walking direction. The user moves from the initial starting point towards the corner of the room (a). In the Follow mode (b), the window moves alongside the user but disregards any rotation. In the Rotation and Follow mode (c), the windows maintain their relative position and orientation with respect to the viewing direction. The Attraction mode (d) shifts and aligns the window with the wall facing the user. With Auto-Centering mode, the window adjusts to the user's body orientation and position (e and f) [Lages and Bowman, 2019]

- **Semantics**: Semantic refers to the meaningful connection between virtual elements and physical objects. Cheng et al. [2021] proposed an optimisation based on semantic associations between virtual interface elements and physical objects. The authors suggested three techniques for semantic placement: anchor, avoid, and replicate (Figure 2.24). The study results indicate that the SemanticAdapt decreased the number of manual interactions in comparison to the baseline adaptation method (fitting usable space of the environment) and was preferred in subjective ratings. In addition, Qian et al. [2022] 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. Their study revealed the high performance and overall usability of the prototype.

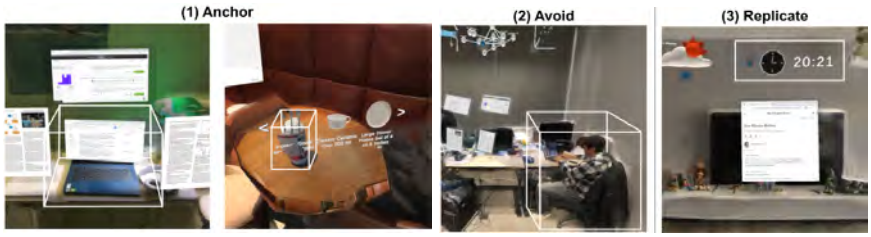


Figure 2.24: Example of semantic constraint: (1) two examples of anchoring. The left image in (1) depicts the utilisation of a laptop computer as a general anchor, while the right image in (1) demonstrates the placement of a kitchen utensil shopping application alongside a physical coffee cup as a semantic anchor. (2) This is an example of avoidance where the user intentionally distantiates the virtual interface elements from their colleague. (3) The replicate behaviour is demonstrated by the user placing a virtual time widget where they would expect to see a physical clock [Cheng et al., 2021]

However, these approaches are limited in two aspects. First, most of these approaches lack user control over the resulting placement optimisation, even though adding interaction into optimisation systems (i.e. human-in-the-loop) has been shown to be beneficial and appreciated by users in other contexts [Liu et al., 2020b]. This control enables users to refine the model to produce desired or acceptable organisations 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, 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.

Summary of Layout Optimisation

Previous works introduced multiple constraint-based approaches to optimise content placement in pervasive interfaces automatically. However, these approaches did not address the challenges of layout optimisation in freeform interfaces. Moreover, the proposed solutions have been tested in isolation from each other and lack any form of user control over the resulting placement optimisation. To sum up, there is a need for intermediate approaches in pervasive context combining user input and automatic layout adaptation.

2.5 *Interaction in AR/VR*

In this section, we focus on interaction for immersive augmented reality environments. We first present previous approaches to spatial manipulation in immersive environments. We then focus on a particular interaction modality, hand gestures, which offers a holistic, fluid [Elmqvist et al., 2011] and intuitive [Jacob et al., 2008] interaction. We review previous works on hand gestures allowing multiple degrees of freedom or conveying meaning through different gestures.

2.5.1 *Spatial Manipulation*

The spatial manipulation of information spaces in augmented reality environments can be classified according to three dimensions [Ens et al., 2014a]: 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 [Xu et al., 2018, Lubos et al., 2016], near the user [Lee et al., 2021b, Yanagihara et al., 2019], or far away [Liu et al., 2020a, Satriadi et al., 2020]. The input mode can be direct (e.g. direct touch) or indirect (e.g. cursors or ray-casting). Most of the time, direct input [Yanagihara et al., 2019, Lubos et al., 2016] is used for near and on-body content, whereas indirect input [Satriadi et al., 2020, Luo et al., 2022] 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. This work focuses on a subpart of the spatial manipulation classification: near and far proximity, indirect input mode and tangible content. Namely, we explore interaction using ray-casting, where virtual content is placed near or far from the user and is mapped on the environment.

2.5.2 *Hand Gestures in AR/VR Applications*

To propose a holistic, fluid [Elmqvist et al., 2011] and intuitive [Jacob et al., 2008] interaction, we must consider an appropriate input method allowing the user to author the optimisation constraints. Hand gestures are

Figure 2.25: Example of spatial interaction: The user observes a layout of a spherical cap with a large central overview [Satriadi et al., 2020]

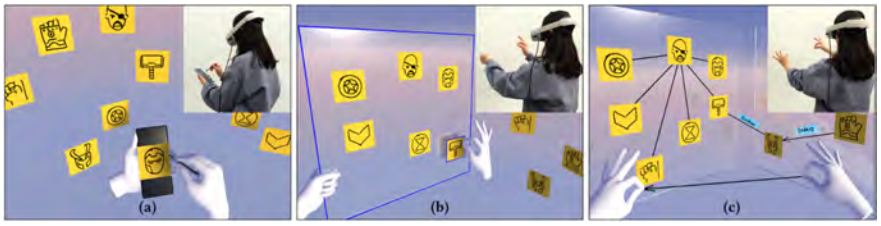


Figure 2.26: In Post-Post-it [Lee et al., 2021b], the user generates ideas for problem-solving: (a) writes on Post-it notes using a motion-tracked smartphone and a stylus, (b) employs realistic one- and two-hand gestures to arrange individual notes and layers of notes at desired positions and orientations in 3D space, and (c) establishes relationships between notes with links to express connections

interesting candidates as they remove the need for any input device, offer multiple degrees of freedom, and can provide semantic meaning through different gestures. Following the use of gestures for surface computing [Wobbrock et al., 2009, Morris et al., 2010], hand gestures have already been successfully used for authoring AR and VR applications [Piumsomboon et al., 2013]. For instance, Arora et al. [2019] explored using 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 adopts a coarse-to-fine workflow to specify rough scenes before focusing on details. Moreover, the authors observed that there are enough distinguishable gestures (Figure 2.27) that do not require expensive hardware for their recognition. In addition, the authors suggest natural interaction with real-world objects in immersive environments to obtain expressive simulation.

Wang et al. [2021] propose an authoring tool, GesturAR, supporting users in 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, such as, for instance, the creation of interactive objects. Yan et al. [2018] demonstrated that object-gesture mappings in virtual reality are highly intuitive, favouring gesture discoverability and memorisation. In their research, the authors explored that the usage, shape and size are important factors when mapping object to gesture. Such object-gesture mapping could be useful in various applications, such as cooking teaching apps or games, to retrieve objects while avoiding the use

of menus.

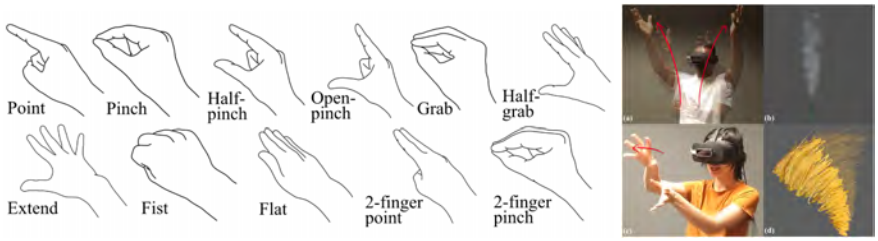


Figure 2.27: Example of hand gestures [Arora et al., 2019]

Summary of Interaction in AR/VR

Spatial interaction using hand gestures has been proven to be useful in many AR application scenarios. Hence, in our work, we decide to apply hand gestures to perform spatial layout adjustment using constraints. However, it is unclear which gestures should be used to define each constraint and how to come up with an entire gesture set with minimal or no gesture delimiter to ensure fluid interaction.

2.6 *Conclusion*

This chapter provided an overview of related work on pervasive freeform user interfaces and their fundamental challenges in terms of content organisation and adaptation. Reviewing the approaches and limitations mentioned earlier helped us to identify the challenges in layout organisation in pervasive interfaces.

Previous approaches of layout adaptation had the limitation of not preserving content visibility or layout properties when dynamically adapting user interfaces to freeform display areas. To address these limitations, we introduce our work Dynamic Decals [Niyazov et al., 2021] - a novel approach to decomposing interfaces into freeform widgets and applying dynamic content placement optimisation using constraints. We describe our approach in detail in Section 3.

Moreover, previous approaches to layout optimisation in augmented reality interfaces lacked user control over the automatic placement of the content. Besides, the proposed constraints were tested in isolation from each other. Our second work, User-driven Constraints [Niyazov et al., 2023], brings interactivity into automatic layout optimisation in AR using hand gestures. We detail our approach in Section 1.7.

3

Dynamic Decals

Our first work answers the following research questions:

- *RQ1 - How to adapt the interface to freeform display areas (outer shape) and the presence of physical objects that can occlude the interface (inner holes)?*
- *RQ2- How to dynamically adapt the interface layout due to the presence of physical objects?*

In this chapter, we present Dynamic Decals, denoted *decal*s subsequently, a novel type of deformable UI widget that is an outcome of answering the aforementioned research questions. This concept is inspired by previous work in computer graphics introducing repetitive patterns composing 2D textures [de Groot et al., 2014]. As illustrated in Figure 3.1, in our approach, all the elements of the graphical interface are represented as decals. Decals can be automatically deformed when colliding with other decals, with physical objects or with the boundaries of the display window. This shape deformation allows the interface content to become freeform and maximize space occupation. Furthermore, our approach allows us to define the position of the group of decals composing the interface to respect certain layout properties, such as alignment. This content repositioning at the decal level preserves content visibility and readability.

In this work, we focus on the research challenges derived from our initial research questions. Elaborating on RQ1 and RQ2 to dynamically adapt the interface to the freeform display areas and the presence of physical objects led us to the following challenges: (1) how to define the decal deformation according to its initial shape and content, and (2) how to define the placement constraints that rule the behaviour of an interface made of several decals.

Our approach is validated by a succession of three user studies showing that Dynamic Decals is an aesthetically pleasing interface that preserves visibility, layout and aesthetic properties.

Figure 3.1: In this work, we model pervasive freeform interfaces as dynamic arrangements of deformable U Iwidgets, called Dynamic Decals. (a) Our approach is fully dynamic, and it automatically adapts the placement and shape of UI widgets w.r.t. (b) user interaction and (c) changes in the screen shape. The relations between the decals are modelled as constraints on their placement and are modified and optimized interactively to (d) enforce specific relations (e.g., alignment) or (e) to adapt the placement to a complex display setup. We model decals as deformable objects (f) that are constrained by the screen boundaries and surrounding decals.

3.1 *Usage Scenario*

Before detailing our contribution, we describe the problems we intend to solve with a usage scenario. We illustrate them in Figure 3.2

Bob is a student at a smart campus, where classrooms have been augmented by projecting interfaces on tables. As the HCI class starts, the system projects the file navigation showing a grid of folders. The table is circular, and the projected interface is square, hence not fully exploiting the available display space (1-space coverage). The teacher starts providing the session instructions, and Bob needs to take some notes and opens his notebook, covering a part of the interface. Then Bob needs to open a specific folder, which is hidden by his notebook (2-object occlusion). He moves the book and selects the folder to open. The folder contains a set of

images. Bob decides to rescale his window to avoid any object occlusion. The window now covers only half of the table, which leaves large parts of the table unused and makes image preview icons too small (3-content visibility). Students need to carry out a 5 minutes brainstorming session using the Post-it application. Bob starts adding and linking post-its. He needs more space, so he decides to enlarge the projected window. Then he needs to manually displace the post-its to the available spaces that are not covered by physical objects, which is quite tedious (4-content positioning).

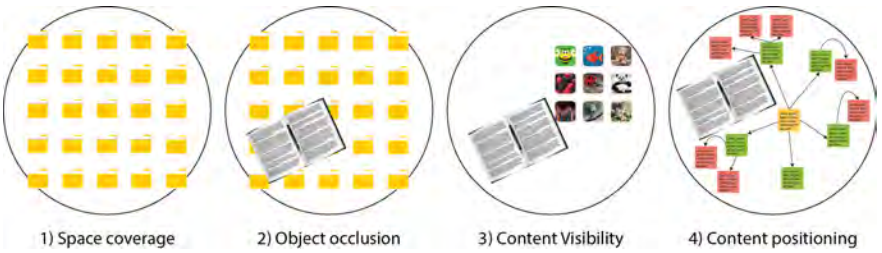


Figure 3.2: Pervasive access to contextual data leads to challenges in (1) efficient space coverage, (2) handling object occlusion, (3) preserving content visibility and (4) content positioning

3.2 *Approach Overview and Contributions*

To solve these challenges and achieve dynamic layout optimisation, we divide the interface into graphical elements (decals) and cast our problem as an interactive dynamic placement problem. Our approach is composed of three main contributions: (1) the definition of a novel type of deformable GUI element, the Dynamic Decals (denoted *decals*); (2) a dynamic layout optimisation model, which moves the decals according to predefined constraints and user interaction; and (3) a deformation model which changes the shapes of colliding decals.

To validate our approach, we conducted three studies. In our first study (Section 3.7), we evaluated the decal shapes and deformers. In our second study (Section 3.8), we validated our constraints and deformation using automatic testing. Finally, we conducted a third study (Section 3.9) to collect user feedback regarding aesthetics, content visibility and usability of our system.

The approach of decomposing the interface into multiple small graphical elements is well adapted to pervasive scenarios applications, such as post-its for brainstorming, gallery of images, or grids of files and folders (see Illustrative Applications Section 3.10.2 in the Discussion). In our work, we focus on such scenarios, although we further discuss how our approach could be extended to consider more complex interface layouts in Future Works (Chapter 5).

3.3 *Layout Constraint Properties*

In this section, we introduce our formulation for the interactive layout adaptation by decomposing the interface into multiple small graphical elements. We propose to optimise the position of the decals according to desired layout properties, e.g. spacing and alignment. Each time the display property changes (e.g., the shape of the display changes, a physical object is moved on top of the display area), our system updates the decal positions.

Our contribution is to extend the use of a constraint-based interactive palette exploration system that provides interactive colour adjustments [Mellado et al., 2017b] (Figure 3.3). The authors analysed a graph-based colour palette defining colours as nodes and the edges as *arbitrary* constraints. The constraints are expressed as a cost function, which is minimized in the least squares sense to find a compromise between them. This enables the design of new constraints specific to GUI layout optimisation. Also, this approach allows the definition of a gamut constraint to force the decals to stay within the display area, similar to penalising colours for remaining in the gamut.

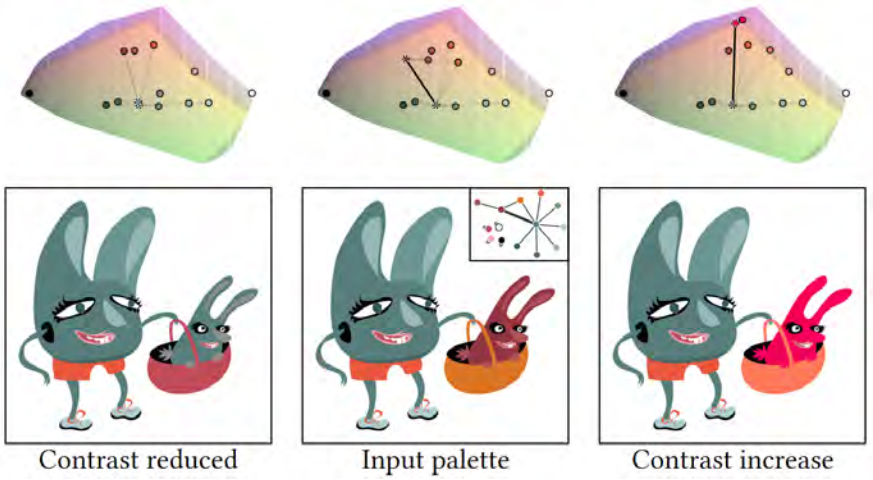


Figure 3.3: Original image (middle), reduced colour contrast (left), increased colour contrast(right). The first row represents a gamut with a graph emphasising colour constraints. [Mellado et al., 2017b]

Following similar notations to Mellado et al. [2017b], we denote $\mathbf{X} = \{\mathbf{x}_i \in \mathcal{S}, i = 1..k_d\}$ the set of decal positions, $\mathcal{S}$ the current display area, and $\Pi(\mathbf{X})$ the *conformity*, which measures how much the decal positions $\mathbf{X}$ respect the desired properties. The optimized positions of the decals $\mathbf{X}^*$ are computed as follows:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}}\, \Pi(\mathbf{X}), \quad \text{with} \quad \Pi^X = \Pi^X_{\mathcal{S}}(\mathbf{X}) + \Pi_{\text{constr}}(\mathbf{X}). \tag{3.1}$$

The goal of the first term is to penalize decals lying outside of the display area, and the goal of the second term is to penalize decals that do not

respect prescribed constraints. They are defined as follows:

$$\Pi_{\mathcal{S}}(\mathbf{X}) = \sum_{i=1}^{k_d} c_{\text{gam}}(\mathbf{x}_i, r_i)^2, \quad \text{with} \quad \Pi_{\text{constr}}(\mathbf{X}) = \sum_{j=1}^{k_c} \left(c_j(\mathbf{X}_j)\right)^2. \tag{3.2}$$

We now present the new constraints we have defined to preserve the desired GUI layout properties: GUI content visibility and layout simplicity, which refers to the number of lines and columns on the layout grid (i.e. if items are aligned, the layout is simpler), and content grouping. We designed these constraints empirically, following the layout simplicity guidelines introduced by Ngo et al. [2003], Galitz [2007]. In these constraints, coordinates and distances are expressed in pixels.

Gamut constraint

The goal of this constraint is to force the decals to stay within the display area (i.e. the gamut). A decal having a radius $r_i = n$ pixels is penalized if it lays *inside* the display area but at less than n pixels from the boundary (see Figure 3.4).

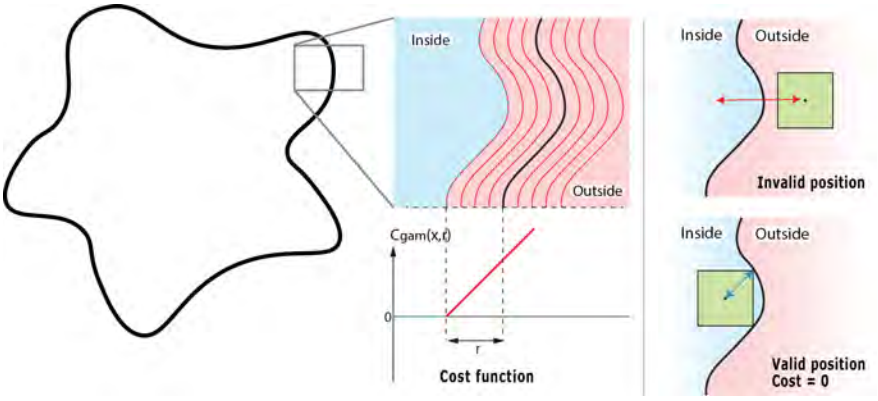


Figure 3.4: Cost function of a gamut constraint and value of the cost according to decal position

In order to ensure fast evaluation with dynamic display area and decals with different radii, we compute a signed distance field $d_{\mathcal{S}}(\mathbf{x})$ to the display area boundary (negative inside, positive outside) and define the gamut constraint $c_{\text{gam}}(\mathbf{x}, r)$ as follows:

$$\begin{aligned} c_{\text{gam}}(\mathbf{x}, r) &= \begin{cases} e_{\text{step}} + d_{\mathcal{S}}(\mathbf{x}) - r & \text{if } d_{\mathcal{S}}(\mathbf{x}) - r > 0 \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \\ d_{\mathcal{S}}(\mathbf{x}) &= \begin{cases} \quad |\mathbf{x} - \text{proj}_{\mathcal{S}}(\mathbf{x})|_2 & \text{if } \mathbf{x} \notin \mathcal{S} \\ - \quad |\mathbf{x} - \text{proj}_{\mathcal{S}}(\mathbf{x})|_2 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.3}$$

The operator $\text{proj}_{\mathcal{S}}(\mathbf{x})$ projects $\mathbf{x}$ on the boundary of the display area $\mathcal{S}$. e_{step} is a constant error term added to out-of-gamut positions, we used $e_{\text{step}} = 10$ for all our experiments.

Distance constraints

We propose two distance constraints to (1) prevent decals' overlap to preserve content visibility and (2) favour decals' grouping. The **minimum distance** (see Figure 3.5) constraint penalizes overlapping decals. In order to ensure fast evaluation of this constraint, we assume, without loss of generality, that the decal's shape can be approximated by axis-aligned square boxes of size r_i. Thus, overlap detection and cost can be efficiently computed using the L_1 distance between the decal's centre:

$$c_{\text{mind}}(\mathbf{x}_a, \mathbf{x}_b) = min\,(0, |\mathbf{x}_a - \mathbf{x}_b|_1 - (r_a + r_b)) \tag{3.4}$$

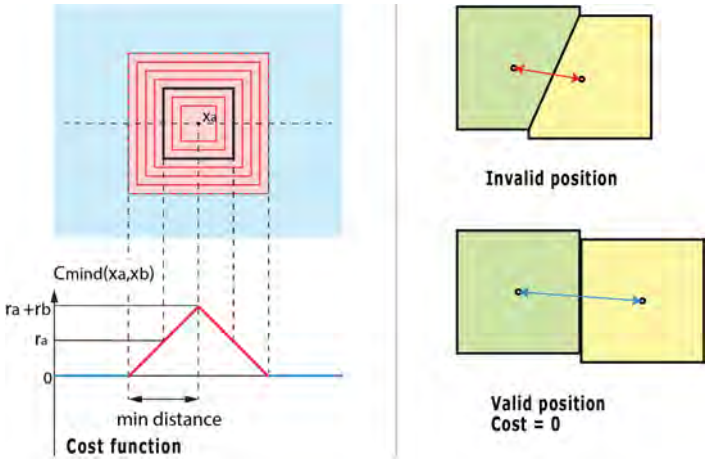


Figure 3.5: Cost function of a minimum distance constraint and value of the cost according to decal position

The **maximum distance** constraint ensures that decals belonging to the same group remain in close proximity to each other. In contrast to c_{mind}, we look for proximity and do not want to favour distances in x and y directions (see Figure 3.6). Thus, we penalize decals when their L2 distance goes higher than a given threshold $d_{\max}$:

$$c_{\text{maxd}}(\mathbf{x}_a, \mathbf{x}_b) = max\,(0, |\mathbf{x}_a - \mathbf{x}_b|_2 - d_{\max}) \tag{3.5}$$

Alignment

The anchor line constraint ensures that decals belonging to the same vertical/horizontal line remain aligned even when one of the decals moves (see Figure 3.7). The cost is computed w.r.t. the distance between the anchor line l and each decal $\mathbf{x}_k$ belonging to a group. Any group of decals can be associated with multiple lines, either vertical, horizontal or both (e.g. to form a grid). Depending on the context, the position of the line can be either fixed (i.e. coordinates are defined when creating the interface) or updated when the display properties change.

The cost is computed w.r.t. the distance between the anchor line l and each decal $\mathbf{x}_k$ belonging to a group denoted $\mathbf{X}$, as:

$$c_{\text{anchor}}(\mathbf{X}) = \sum_k \left|\text{proj}_l\,(\mathbf{X}_k) - \mathbf{X}_k\right|_2 \tag{3.6}$$

Figure 3.6: ost function of a maximum distance constraint and value of the cost according to decal position

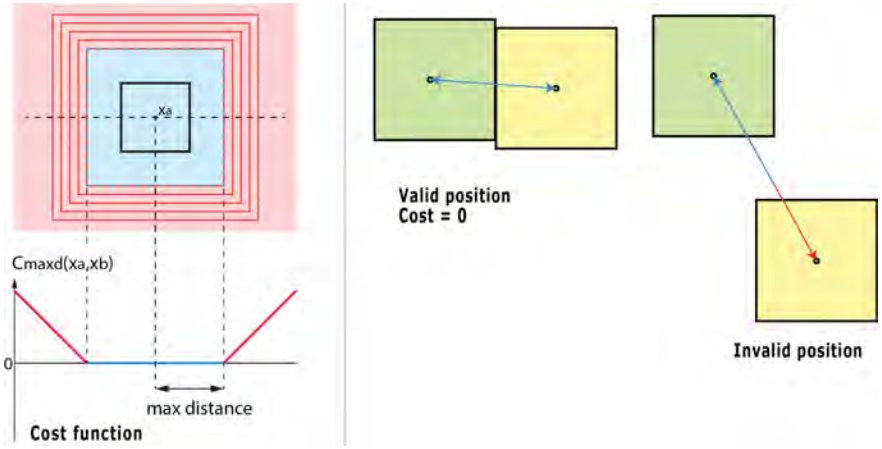
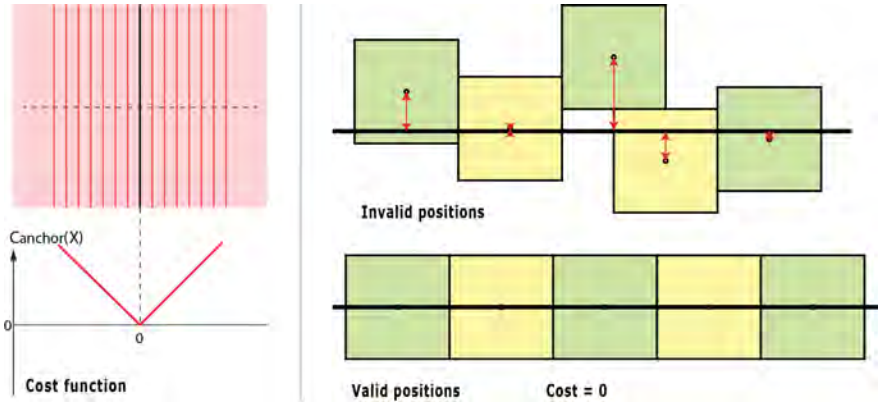


Figure 3.7: Cost function of an alignment constraint and value of the cost according to decal position



3.4 *Collision detection and Contact Deformation Models*

In the previous section, we described interactive layout adaptation when the properties of the interface are changing. However, the placement of graphical elements introduces new challenges of decal adaptation that cannot be fully optimised using constraints. For instance, with the presence of physical objects, the display area becomes smaller, and the decal position cannot be further constrained to penalise overlapping conditions.

A preliminary step to address this challenge is to detect collisions between interface elements. While not very challenging in the case of rectangular axis aligned elements, this detection becomes tedious on freeform objects [Jiménez et al., 2001, Man et al., 2014]. A standard solution is to accelerate the collision query using a bounding objects hierarchy [Gottschalk et al., 1996, Ericson, 2004, He and van den Berg, 2014], eventually optimized using probabilistic computations [Park et al., 2017]. Other solutions first

query simple proxies approximating the complex shapes [Hoff et al., 2001] or use polar diagrams in which the collision detection is based on angle processing [Ortega and Feito, 2005].

When colliding, the shape of elements is deformed with physical behaviours, using either explicit force modeling [Brenner and Carstensen, 2017] or constraint-based approximations [Müller et al., 2007, Bender et al., 2017]. These approaches remain computationally intensive, and physical parameters are difficult to parameterize. Alternatively, handle-based geometric approaches enable user-defined free-form deformations [Jacobson et al., 2014]; however, the coupling with contact detection remains intricate.

Depending on the deformation models, deformation applies either on the whole area of the element [Brenner and Carstensen, 2017, Müller et al., 2007] or only on its boundaries [Brunel et al., 2020]. In the first case, the element content is directly deformed by the model. In the second case, a *mapping* between the original and the deformed shape areas is estimated to maintain some geometric properties such as preserving angles (conformal mapping), areas (equiareal mapping), or some mapping properties such as null Laplacian (harmonic mapping) [Hormann et al., 2007]. In this work, we target interfaces composed of elements of different natures (e.g., images, icons, text boxes, post-its), which might all require a dedicated deformation, complex to design by acting on physical or mapping deformation parameters. In addition, our elements may dynamically change topology (i.e. genus), which is an important issue for such approaches.

Another solution comes from decal-based 3D object surface texturing [Lefebvre et al., 2005, Schmidt et al., 2006]. These methods populate a surface with large sets of small tiles, the same way icons and content would populate a screen. Recently, de Groot et al. [2014] defined a new model called Implicit Decals, which enables the automatic generation of colliding deformations between decals without requiring any collision detection procedure. By explicitly modelling the area of a decal as a 2D field function [Bloomenthal and Wyvill, 1997], Implicit Decals deform both the boundaries and the objects' areas. While de Groot et al. [2014] only introduces a single deformation operator, both the boundary and the inner area deformation may be controlled using other state-of-the-art free-form composition operators [Barthe et al., 2003, Gourmel et al., 2013, Angles et al., 2017], by relying only on user-defined geometric parameters. This approach also seamlessly handles freeform surfaces with varying topology, which would allow our content to naturally adapt to freeform screens of arbitrary genera.

3.5 *Decal Properties*

In this section, we first present a technical background of Dynamic Decals by extending Implicit Decals [de Groot et al., 2014]. Furthermore, we present our decals and introduce their dedicated shapes and a different deformation model.

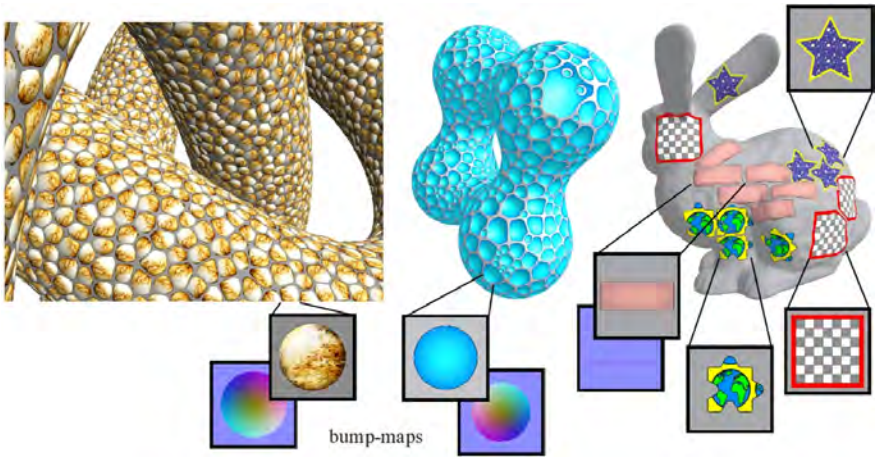


Figure 3.8: The texture is mapped to multiple forms using Implicit Decals [de Groot et al., 2014]

3.5.1 *Technical Background on Decals*

Decals (i.e. small tiles used to texture the surface of a freeform 3D object) are introduced with an implicit formulation by de Groot et al. [2014] and denoted as Implicit Decals. An Implicit Decal D_i is defined by a field function $f_i(p) : \mathbb{R}^2 \rightarrow [0,1]$ whose value is 1 at the decal centre, 0 at all points p along its influence limit and $\frac{1}{2}$ at points on the decal boundary. As illustrated in Figure 3.9-a, in a decal, the content is mapped in the area where the field value is in the range $[\frac{1}{2}, 1]$ while the outer decal values in the range $[0, \frac{1}{2}]$ are required for implicit deformation formulations [Wyvill et al., 1986, Bloomenthal and Wyvill, 1997, Wyvill et al., 1999, Barthe et al., 2004].

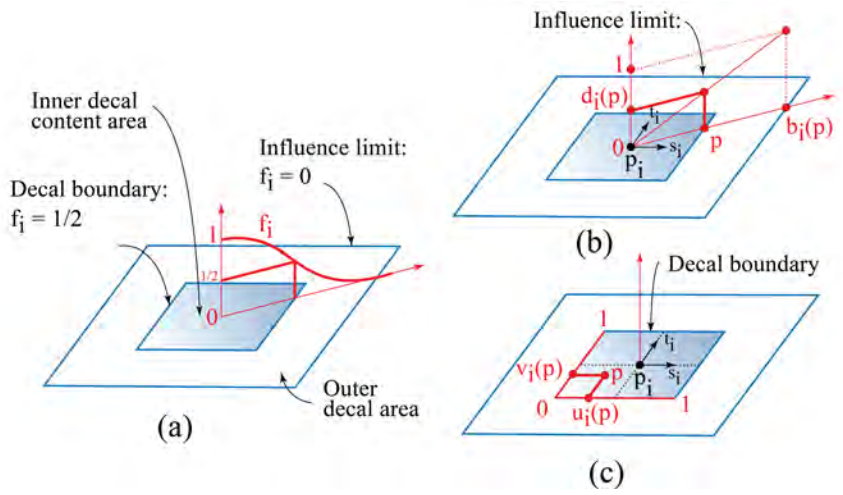


Figure 3.9: Field function of a square decal. (a) Content is mapped in the decal-hatched area. (b) Distance field $d_i(p)$ over a decal. $b_i(p)$ computes the Euclidean distance of the decal influence limit in the direction of p. (c) Decal parameterization in the decal display area.

A decal position is set by its center p_i, where $f_i(p_i) = 1$ and its orientation is given by a local frame $(\vec{s_i}, \vec{t_i})$ (this frame is illustrated in Figures 3.9-b

and 3.9-c). In the implicit decal formulation, the field function f_i is defined by composing two components. A distance field $d_i : \mathbb{R}^2 \to [0,1]$, providing a distance, eventually anisotropic (i.e. varying differently in different directions), from any point p in space and the decal centre p_i, and a compactly supported fall off function $g : [0,1] \to [0,1]$ such that $f_i(p) = g(d_i(p))$ exhibits the aforementioned properties illustrated in Figure 3.9-a. In de Groot et al. [2014] implicit decals, they are defined as follows:

$$d_i(p) = \frac{||p - p_i||}{b_i(p)}, \text{ and } g(d) = \begin{cases} (1-d^2)^3 & \text{if } d \leq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3.7}$$

where the function $b_i(p)$ defines the decal shape by computing the Euclidean distance from p_i and the decal influence limit in the radial direction $\overrightarrow{p_i p}$ as illustrated in Figure 3.9-b in the case of an anysotropic function b_i defining a square decal. For a circular decal of radius R_i, we would just set $b_i(p) = R_i = C^{te}\ \forall p$ (see Figure 3.10-c).

A fundamental advantage of the field function formulation is to enable the automatic generation of contact deformations between n decals by combining their field functions f_i with a composition operator $l : [0,1]^n \to [0,1]$ and defining a unique field function $F : \mathbb{R}^2 \to [0,1]$ as $F(p) = l(f_1(p), .., f_n(p))$, which includes all decals deformed. De Groot's contact formulation is defined as:

$$F(p) = \frac{1}{2} + \left(f_k(p) - \frac{1}{2}\right) \prod_{j \neq k} h(f_j(p), f_k(p)) \text{ where } k = \underset{i}{\operatorname{argmax}}\,(f_i(p)) \tag{3.8}$$

and

$$h(x,y) = \begin{cases} 1 - (\frac{x+y-1}{2y-1})^{\frac{1}{1-y}} & \text{if } x + y \geq 1 \\ 1 & \text{otherwise.} \end{cases} \tag{3.9}$$

To summarize, a decal D_i is controlled by its center position p_i, its local frame $(\vec{s_i}, \vec{t_i})$ and its shape function $b_i(p)$.

Finally, the key idea of the implicit decal representation is to define its parameterization $(u_i(p), v_i(p))$ using the inverse function $g^{-1}(F(p))$. It returns the distance $d_i(p)$ in an isolated decal D_i and automatically transfers the decal deformations on the parameterization when they collide. De Groot et al. introduced the parameterization in polar coordinates. In our case, the cartesian (u, v) coordinates defined in $[0,1]^2$ are more convenient and we derive them as:

$$\begin{cases} u_i(p) = \dfrac{1}{2}\left(\dfrac{\tilde{b_i}(p).s_i}{b_i(p_i+\vec{s_i}).g^{-1}\left(\frac{1}{2}\right)}+1\right) \\ v_i(p) = \dfrac{1}{2}\left(\dfrac{\tilde{b_i}(p).t_i}{b_i(p_i+\vec{t_i}).g^{-1}\left(\frac{1}{2}\right)}+1\right) \end{cases} \text{with } \tilde{b_i}(p) = g^{-1}(F(p)).bi(p).\frac{p-p_i}{||p-p_i||} . \tag{3.10}$$

3.5.2 *Our Decals*

Following this definition of implicit decals, we first present functions b_i producing our decal shapes. We then introduce a new technical contribution: the use of field function deformers with their formulation. In the literature, almost all the field function interactions are modelled with composition operators [Sabin, 1968, Ricci, 1973, Wyvill et al., 1986, Pasko et al., 1995, Bernhardt et al., 2010, Gourmel et al., 2013] that combine the field functions hierarchically in a tree in which each node is a field function representing the composition of its children [Wyvill et al., 1999]. We propose to avoid this tree structure (also used in implicit decals [de Groot et al., 2014]) in which the root is a single field function F representing the result of all the successive deformations of each leaf field function f_i. Following the seminal work of Cani [1993] on contact deformations modelling with field functions, we rather deform each field function f_i as a new field function f_i^1. Each additional deformation applied on a field function thus transforms f_i^j in f_i^{j+1}. When drawing decals, this avoids the need to evaluate the whole root function F (including its tree structure) while determining from which deformed function f_i both texture and shape function b_i are to be used. With deformers, each decal is defined by its own field function, and thus, it also includes its own deformed parameterization with which its own content is mapped. This is both more efficient and not subject to any indeterminacy when complex deformations are used.

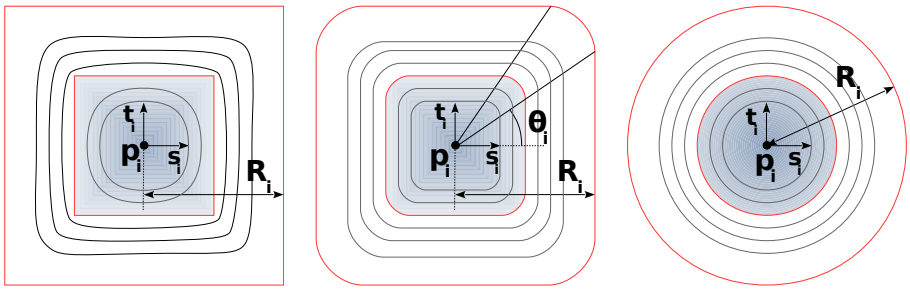
This is implemented by replacing $g^{-1}(F(p))$ by $g^{-1}(f_i^k(p))$ in Equation 3.10, which yields the following formulation for $\tilde{b_i}$(p):

$$\tilde{b_i}(p) = g^{-1}(f_i^k(p)).b_i(p).\frac{p-p_i}{||p-p_i||} , \tag{3.11}$$

where k is the index of the last deformation applied on the decal on which the parameterization is computed.

3.5.3 *Decal Shape*

We define three different decal shapes corresponding to the most common element shapes in GUIs. We present below the shape functions b_i of a square decal, a square decal with rounded corners and a circular decal (Figure 3.10)



(a) (b) (c)

Figure 3.10: (a) Square, (b) rounded square and (c) a circle decal shapes. In red, iso-lines $f_i = \frac{1}{2}$ and $f_i = 0$ define the decal boundary and its limit of influence, respectively.

A square decal is defined as a square at the decal boundary and at its influence limit (see Figure 3.9). Squares are interpolated with a circle to smooth the in-between field (see Figure 3.10-a) and thus the contact deformations. A square of size R_i (i.e. an edge length of $2R_i$), is defined by the shape function b_i^s as follows:

$$b_i^s(p) = t^2 \bar{b}_i^s(p) + (1-t^2)R_i \quad , \quad \text{with} \quad t = \begin{cases} \frac{s}{r_i} & \text{if } s \leq r_i \\ \frac{R_i - s}{R_i - r_i} & \text{if } r_i < s \leq \frac{r_i + R_i}{2} \\ \frac{r_i - s}{r_i - R_i} & \text{if } \frac{r_i + R_i}{2} < s \leq R_i \end{cases} \tag{3.12}$$

where

$$\begin{aligned} r_i &= g^{-1}\left(\tfrac{1}{2}\right) \quad , \quad s = \max(|\vec{s_i}.(p-p_i)|, |\vec{t_i}.(p-p_i)|) \quad , \text{ and} \\ \bar{b}_i^s(p) &= \frac{R_i \, ||p-p_i|| }{s} \quad . \end{aligned} \tag{3.13}$$

A square with rounded corners decal is represented by squares whose corners are defined by a quarter of a circle. For a square of size R_i rounded at an angle θ_i (see Figure 3.10-b), the shape function b_i^{rs} is defined as follows. We first define:

$$s_p = |\vec{s_i}.(p-p_i)| \quad \text{and} \quad t_p = |\vec{t_i}.(p-p_i)| \quad . \tag{3.14}$$

Then, if $(t_p \leq s_p \tan\theta_i)$ or $(s_p \leq t_p \tan\theta_i)$, the decal is a square and:

$$b_i^{rs}(p) = \frac{R_i \, ||p-p_i|| }{\max(s_p, t_p)} \quad , \tag{3.15}$$

Otherwise, we compute the intersection of a ray starting from p_i in the direction of p and the arc of the circle defining the rounded corner. To do so, we first compute the:

$$\vec{r} = \frac{1}{||p-p_i|| } \begin{pmatrix} s_p \\ t_p \end{pmatrix} \quad \text{and} \quad \vec{o} \begin{pmatrix} R_i \tan\theta_i \\ R_i \tan\theta_i \end{pmatrix} \quad , \tag{3.16}$$

then:

$$a = \vec{r}^2 \ , \ b = -2\vec{o}.\vec{r} \ , \ c = \vec{o}^2 - (R_i(1 - \tan\theta_i))^2 \ , \tag{3.17}$$

and finally:

$$b_i^{rs}(p) = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \ . \tag{3.18}$$

A circular decal is easily defined with a constant shape function set with its size: $b_i^c(p) = R_i \ \forall \ p$, as depicted in Figure 3.10-c.

3.5.4 Deformers

When modelling the contact interaction of decals, there are two aspects to define: the way the content (i.e. the inside part) deforms and the way the boundary deforms. We achieve such deformation with the help of deformers. A deformer is a function $z : [0,1]^n \rightarrow [0,1]$, which, composed of all field functions in interaction, computes the deformation of a field function f_i as $f_i^1(p) = z(f_i, \ldots, f_j(p), \ldots)$, ($j = 1..n$ and $j \neq i$). Concerning the content, our goal is to preserve its visibility during deformation. For the boundaries, the deformer could tend to preserve its shape to accentuate the separation between decals or, on the contrary, fuse them to indicate that decals belong to the same group of GUI elements (e.g. as toolbar icons).

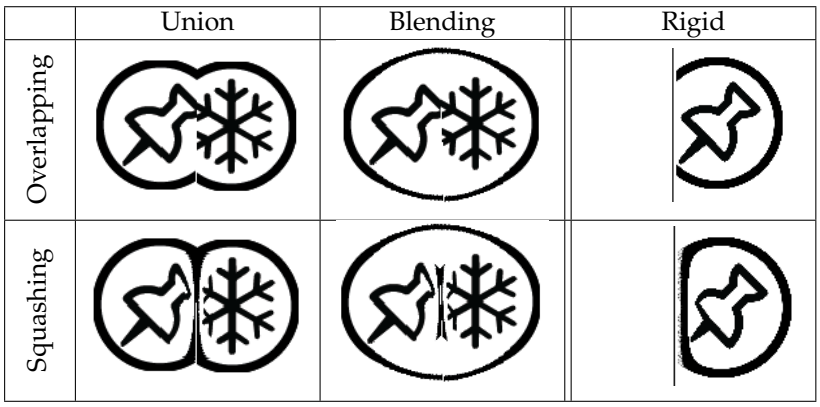


Figure 3.11: The different types of deformers modelling the different contact behaviours for our study. By row, the inside content behaviour (overlapping or squashing) and by column, the decal boundary behaviour (union, blending or rigid).

In this work, we introduce four deformers (Figure 3.11) for modelling UI objects interactions when they collide. They are based on two inside behaviours - squashing or overlapping (Figure 3.11-rows) - and two decal boundary behaviours - union or blending (Figure 3.11-left and middle columns). The special case of the display boundary that remains rigid when a decal collides - i.e. only the decal is deformed - is handled with the same two inside decal behaviours (squashing or overlapping). These two additional deformers, illustrated in Figure 3.11-right, are not part of the study (Section 3.7) as the squashing or overlapping behaviours of the decals are already evaluated.

Inspired by the design of composition operators, we create our deformers following a result-guided procedure [Barthe et al., 2001, 2003, Angles

et al., 2017]. This allows us to select a desired resulting deformation on the field function and to use it to derive the shape of the corresponding deformer directly. We then propose equations generating the corresponding shape for the deformer z.

For all deformer equations, we set k as:

$$k = \arg\max_l (f_l) \ , \ l = 1..n \ . \tag{3.19}$$

The Overlapping-Union deformer is built to keep unchanged the field function f_i outside all other decal boundaries, and where $f_i \geq f_j$ ($j = 1..n$ and $j \neq i$). Where $f_i < f_j$, f_i is set to a value in $[0, \frac{1}{2}]$ to define the outer area of the deformed decal adequately. We thus propose the following formulation:

$$z^{ou}(f_i, \ldots, f_j, \ldots), \ j \neq i, \ = \begin{cases} f_i & \text{if } i = k \text{ or } (f_i \leq \frac{1}{2} \text{ and } f_k \leq \frac{1}{2}) \\ \text{otherwise:} & \\ f_i - f_k + \frac{1}{2} & \text{if } f_i - f_k + \frac{1}{2} > 0 \\ 0 & \text{otherwise .} \end{cases} \tag{3.20}$$

The Squashing-Union deformer is very similar to the overlapping-union (Equation 3.20). The difference is in the inner decal part, where we want to squash the decal content. This is done using an adaptation of de Groot contact composition operator [de Groot et al., 2014] as follows:

$$z^{su}(f_i, \ldots, f_j, \ldots), \ j \neq i, \ = \begin{cases} H(f_i, f_k) & \text{if } i = k \\ f_i & \text{if } f_i \leq \frac{1}{2} \text{ and } f_k \leq \frac{1}{2} \\ \text{otherwise:} & \\ f_i - f_k + \frac{1}{2} & \text{if } f_i - f_k + \frac{1}{2} > 0 \\ 0 & \text{otherwise ,} \end{cases} \tag{3.21}$$

where

$$H(x, y) = \tfrac{1}{2} + (x - \tfrac{1}{2})h(x, y) \ , \tag{3.22}$$

and $h(x, y)$ is defined in Equation 3.9.

The Overlapping-Blending deformer has the same inner behaviour as the overlapping-union (Equation 3.20) except for a small inner band of width e close to the decal boundary that is mapped on the blending boundary. This enables the generation of the black contouring of the blended decals, as can be seen in Figure 3.11-right. Blending is a well know composition operator for combining field functions, and we propose to model it using its simplest formulation, i.e. by summing the combined field functions [Blinn, 1982, Wyvill et al., 1986]. In our case, the blending is only located in the small inner band of width e and in the outside decal

areas. We thus end up with the following formulation of our deformer:

$$z^{ob}(f_i,\ldots,f_j,\ldots),\ j\neq i,\ = \begin{cases} f_i & \text{if } i=k \text{ and } f_i > \frac{1}{2}+e \\ \text{otherwise:} & \\ \sum_{l=1}^{n} f_l & \text{if } \sum_{l=1}^{n} f_l \leq \frac{1}{2}+e \text{ and } i=k \\ \text{otherwise:} & \\ f_i - f_k + \frac{1}{2} & \text{if } 0 < f_i - f_k + \frac{1}{2} < \frac{1}{2} \\ 0 & \text{if } f_i - f_k + \frac{1}{2} \leq 0 \\ \frac{1}{2}+e & \text{otherwise .} \end{cases} \tag{3.23}$$

The Squashing-Blending deformer, follows the formulation of the overlapping-blending deformer (Equation 3.23). We have to add the squashing deformation inside the decal. It is thus defined as follows:

$$z^{sb}(f_i,\ldots,f_j,\ldots),\ j\neq i,\ = \begin{cases} H(f_i,f_k) & \text{if } i=k \text{ and } f_i > \frac{1}{2}+e \\ \text{otherwise:} & \\ \sum_{l=1}^{n} f_l & \text{if } \sum_{l=1}^{n} f_l \leq \frac{1}{2}+e \text{ and } i=k \\ \text{otherwise:} & \\ f_i - f_k + \frac{1}{2} & \text{if } 0 < f_i - f_k + \frac{1}{2} < \frac{1}{2} \\ 0 & \text{if } f_i - f_k + \frac{1}{2} \leq 0 \\ \frac{1}{2}+e & \text{otherwise .} \end{cases} \tag{3.24}$$

where $H(x,y)$ is defined in Equation 3.22.

The Squashing-Rigid deformer is used to squash a decal field f_1 against a rigid display field f_2 boundary. It is thus a binary deformer $z^{sr}(f_1,f_2)$. It is built such that where the decal and the display interpenetrate and outside the display boundary, the outer display is removed from the decal. Inside, an inside squashed field is built along the display boundary. This is implemented as follows:

$$z^{sr}(f_1,f_2) = \begin{cases} f_1 & \text{if } f_1 \leq \frac{1}{2} \text{ and } f_2 \leq \frac{1}{2} \\ 1-f_2 & \text{if } f_1 \geq \frac{1}{2} \text{ and } f_2 \geq \frac{1}{2} \\ \max\left(0, \frac{1}{2} - \sqrt{\left(\frac{1}{2}-f_1\right)^2 + \left(\frac{1}{2}-f_2\right)^2}\right) & \text{if } f_1 < \frac{1}{2} \text{ and } f_2 > \frac{1}{2} \\ (1-t(f_2))f_1 + \frac{t(f_2)}{2} & \text{otherwise ,} \end{cases} \tag{3.25}$$

where

$$t(x) = (2x)^4 \tag{3.26}$$

The Overlapping-Rigid deformer is very similar to the Squashing-Rigid deformer (Equation 3.25). The only difference is that the inner decal content is hidden by the gamut rather than being squashed. Its equation is

thus:

$$z^{or}(f_1, f_2) = \begin{cases} f_1 & \text{if } f_1 \leq \frac{1}{2} \text{ and } f_2 \leq \frac{1}{2} \\ 1 - f_2 & \text{if } f_1 \geq \frac{1}{2} \text{ and } f_2 \geq \frac{1}{2} \\ \max\left(0, \frac{1}{2} - \sqrt{\left(\frac{1}{2} - f_1\right)^2 + \left(\frac{1}{2} - f_2\right)^2}\right) & \text{if } f_1 < \frac{1}{2} \text{ and } f_2 > \frac{1}{2} \\ f_1 & \text{otherwise .} \end{cases} \tag{3.27}$$

To this end, we defined our decals by expanding Implicit Decals and proposing to deform each decal separately. We presented three different decal shapes that can be deformed upon collision with other decals or display boundaries. The deformation is achieved with the help of four deformers that are composed of field functions and applied to the boundaries and internal content of the decals.

3.6 Implementation

To achieve real-time performances, we use the versatile solver proposed by Mellado et al. [2017b], which allows for defining arbitrary constraints between optimized elements. After each layout optimization step, we deform the colliding GUI elements with other elements or with display boundaries.

Decals Each decal content is stored as a 2D texture. Then, each decal field function is precomputed in another 2D texture of the same size. At run time, a parameterization decal texture is used. Before being displayed, deformers of each decal, followed by the parameterization computation, are applied on the precomputed field function texture and stored in the parameterization decal texture. A decal D_i is then rendered, for a given pixel p, by reading the parameterization $(u_i(p), v_i(p))$ in the parameterization texture and fetching the corresponding colour in the content texture. A standard axis aligned 2D bounding box hierarchy [Gottschalk et al., 1996] enables optimized decal collision detection for computing deformations only between potentially colliding decals.

Interactive processing pipeline Our approach is designed to dynamically update a GUI when the display area or its content change: we call such event an *update event*. Our prototype takes as input an augmented GUI, where GUI elements are modelled as decals, and their relations are described as constraints. Each time an update event is triggered, the system starts by updating the gamut: a 2D binary texture (black inside, white outside) caches the gamut of the current frame. Similarly, we compute and store in a 2D texture the associated signed distance field. For the studies, we combine a static binary image representing the display area, modified by adding the footprint of the object at a given location. With this approach, the system can easily handle multiple physical objects since no additional computation is needed to integrate each object. Then the solver is run,

taking as input the decals, their relations, and the gamut distance field. The decals are deformed, and the GUI is rendered using the Qt library. Our prototype achieves an interactive framerate on a standard laptop with a CPU Intel(R) Core(TM) i7-8850H 2.60GHz and 32GB RAM.

Optimization As Mellado et al. [2017b], we minimize the constraint costs with the single thread implementation of the Levenberg-Marquardt (LM) algorithm provided by Eigen [Guennebaud et al., 2010]. As LM is a non-linear algorithm, it takes as input starting decal positions modified to minimize the energy. In order to get stable optimization results, we set the starting positions as the initial decal positions. When a user grabs and moves a decal in the GUI, the position of all the other decals is optimized but not the selected one.

3.7 Study 1: Decal Shape and Deformation

To evaluate the previously presented decal shapes and deformers, we chose to conduct an online pairwise experiment in which participants had to rate pairs of conditions (in our case, pairs of animations showing two decals colliding). Such experiments are typical to gather Quality of Experience (QoE) feedback and have been previously used to compare freeform interface layouts [Serrano et al., 2017].

3.7.1 Study Description

We adopted an approach similar to [Serrano et al., 2017] by conducting an online survey and using the Bradley-Terry-Luce [Bradley and Terry, 1952] mathematical model to analyse the results, as detailed below.

Task

We conducted an online survey, which was divided into two parts. For the first part, the task consisted of comparing pairs of deformers. Each deformer was illustrated using an animated gif showing two decals colliding (see them here). For each pair of deformers, participants had to answer three questions to say which deformer was nicer (i.e. visually pleasing), clearer (i.e. with more visible/readable content), and which one better represented a group of items (association). The first two questions correspond to aesthetic terms proposed in the literature [Lavie and Tractinsky, 2004]. We focused on aesthetics because previous work has shown a strong correlation between aesthetics and usability [Kurosu and Kashimura, 1995]. The last question focused on whether, when two decals collide, the deformation makes it clear that the two decals are grouped since GUI items are usually grouped (such as toolbar items). For each question, the participant could provide three answers: pair A, pair B or both (i.e. equality). For the second part of the survey, the participants were asked to rank the decal shapes in order of preference for each type of content (icon, image and text) and motivate their choice.

Evaluated conditions

We tested the three decal shapes (rectangular, rectangular with rounded borders and circular) and the four deformers (Overlapping-Union, Overlapping-Blending, Squashing-Union and Squashing-Blending, see Figure 3.12). Concerning the decals content, we decided to study three common types of UI content: icons, images or text. Notice that we decided to only study collisions between decals having the same shape and content. As for the text, we used dummy Lorem Ipsum text. There were eleven words in both square and rounded square decal shapes, whereas only nine words fit in circular shape decals having the same font size. Regarding images and icons, we chose them so that the contact behaviour was noticeable enough on decals collision. All deformers were shown using the same collision trajectory; the speed was empirically set to 54 pixels per second, so the deformation was visible but not too fast. The speed of decals in the final application depends on the solver, and we do not apply any speed constraint (e.g. decals can move slower or faster, according to the user's actions). The maximum overlapping area of the decals was around 40%, which is the maximum overlap that would occur in our final system. In total, we generated 36 animations corresponding to 4 deformers x 3 shapes x 3 content types.

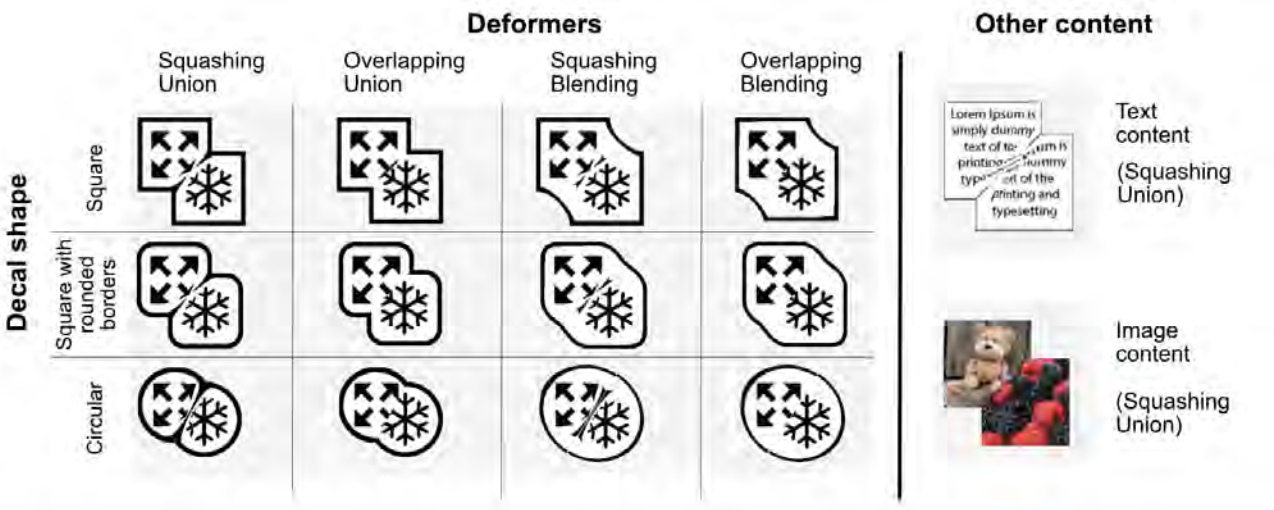


Figure 3.12: In this survey, we evaluated three decal shapes and four deformers (Left, illustrated with the Icon content). We also investigated these conditions with Text and Image content (Right, illustrated with Squashing-Union on a rectangular decal).

Participants

We had 31 complete answers to our study, but we removed results from 4 participants due to their low consistency (Section 3.7.2). The final 27 participants (13 female) were aged 28.1 on average (SD=6.9). 14 of them were university students, while the rest were professional workers from various areas: 3 researchers, three office workers, a teacher, a photograph, an illustrator, a computer scientist, a flight attendant, a surgeon and an

architect.

Study Design

This study followed a 4x3x3 within-subject design, with Deformation operators (Overlapping-Union, Overlapping-Blending, Squashing-Union and Squashing-Blending), Shape (Square, Square with rounded borders or Circular) and Content type (Icon, Image or Text) as factors. For the first part of the study, participants evaluated 54 pairs of animations (6 pairs combining the 4 deformers x 3 content types x 3 shapes). We used a 3x3 Latin square to counterbalance the order of the Shape factor. For the second part of the study, we asked participants to rank the three shapes for each content type.

3.7.2 *Results*

We first report on the consistency of the collected data before describing the results of the pairwise comparisons of the deformers and the shape preference.

Data consistency

Our analysis of the results is based on the three steps described in [Chen et al., 2009], which validate the consistency of the collected data. The first step consists in computing the Transitivity Satisfaction Rate (TSR) to analyse the individual consistency judgement over multiple questions. For instance, if the participant rates A greater than B, and B greater than C, then we expect A to be greater than C. If this inequality is not preserved, the TSR is decreasing. Based on [Chen et al., 2009], we calculated the TSR in Python. After measuring the TSR for each participant, we removed 4 of them due to a low value (between 0.7 and 0.8). For the resting 27 participants, the TSR was above 0.8, meaning that they paid full attention to the study [Chen et al., 2009]. Second, we calculated the group consistency judgement using Kendall's tau coefficient [Chen et al., 2009]. The Kendall tau coefficient was greater than 0.5 for all answers to the 'nicer/clearer' questions, meaning the overall judgement was consistent enough [Chen et al., 2009]. The answers to the 'grouping' question provided a Kendall tau coefficient above 0.5 only for some conditions (shapes having the text as content, as well as circular icons). Therefore, we provide the 'grouping' results only for these conditions.

Significance Codes		
nice ***	clear ***	group ***
nice **	clear **	group **
nice *	clear *	group *
nice	clear	group

	P > 0.05
*	P ≤ 0.05
**	P ≤ 0.01
***	P ≤ 0.001

Figure 3.13: (a) Color encoding of the p-values in the estimate tables for the three metrics: nice, clear and grouping. (b) Numerical ranges of p-values

Finally, we used the Bradley-Terry-Luce model [Bradley and Terry, 1952] to get an "ability" metric for each condition as well as statistical differences

between conditions with p-values. For clarity, we colour code these p-values (Figure 3.13) and illustrate them in detail for Square decal shape, Square decal with rounded borders and decal with Circular shape in Figure 3.14.

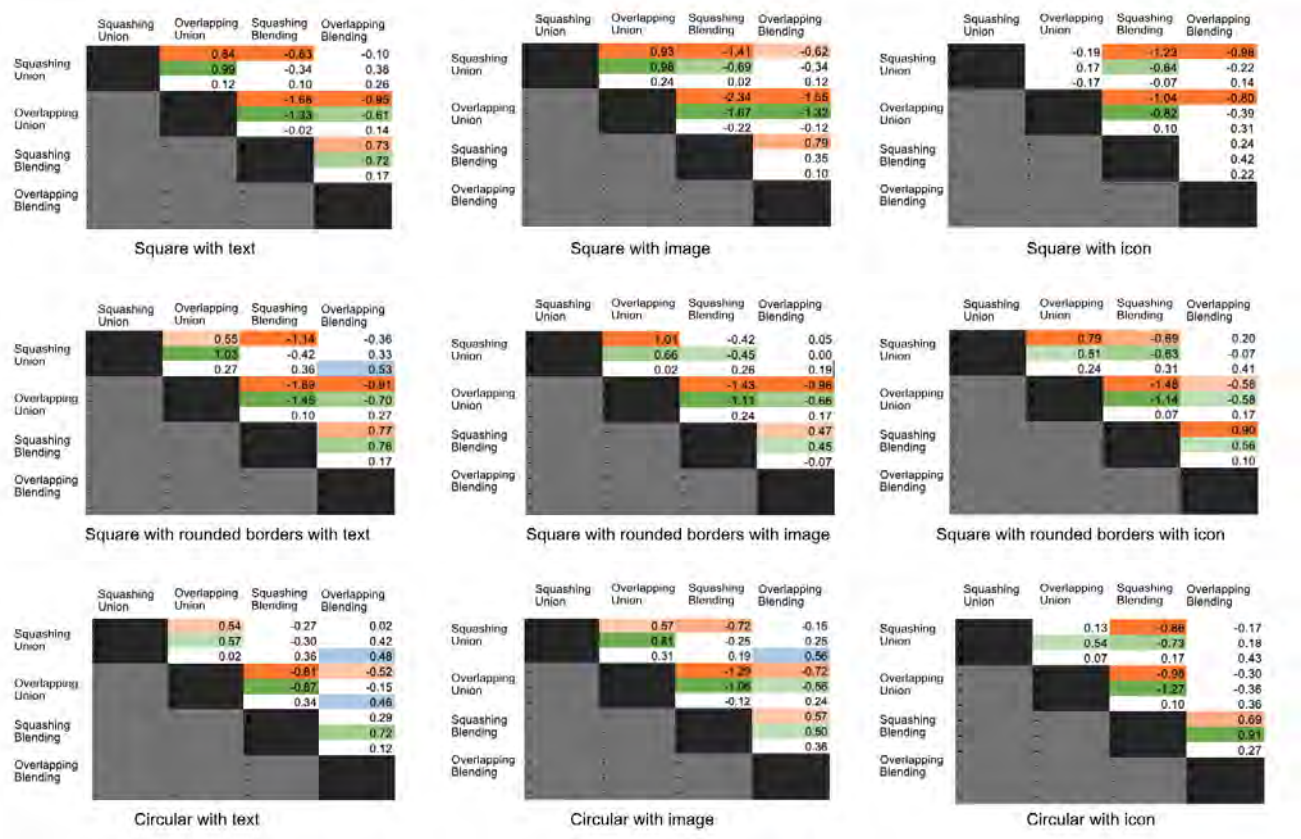


Figure 3.14: Each row illustrates tables with the estimate values for the Square decal shape, square decals with Rounded corners and decals with Circular shape. Significance is colour encoded.

Visibility and aesthetics

Figure 3.15 shows the results representing the Nice and Clear estimate values as computed via the Bradley-Terry-Luce model. For all figures, the Squashing-Union operator was used as a reference. Hence, the estimates for the other deformation operators are provided with respect to Squashing-Union.

Results on the Nice and Clear metrics are consistent across decal shapes and contents. Overall, the Overlapping-Union operator was rated nicer and clearer than the other deformation operators (see Figure 3.15). Squashing-Union and Overlapping-Blending were perceived to be the most clear and nice after Overlapping-Union: we found no statistical difference between Squashing-Union and Overlapping-Blending except for rectangular icons and images, where Squashing-Union was deemed nicer. There were only

two conditions where Overlapping-Union values were not significantly different than Squashing-Union and Overlapping-Blending: for icons presented on rectangular or rounded decals. The Squashing-Blending operator was always rated the least nicer and clearer.

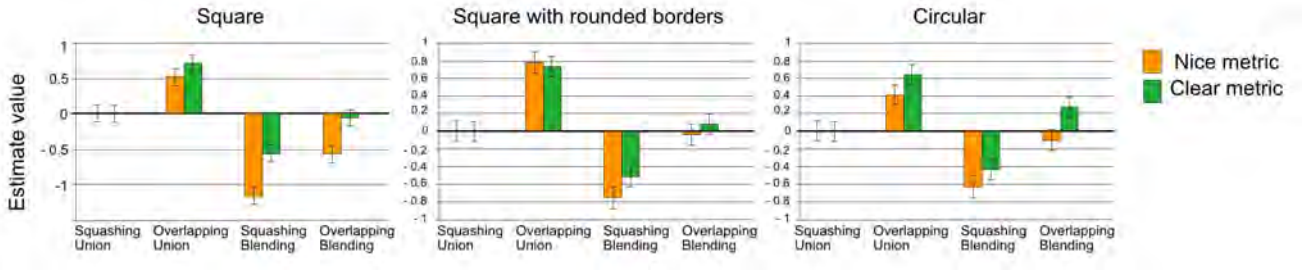


Figure 3.15: Results for the pair comparisons between deformers for each decal shape (the y-axis represents the mean rating with respect to Squashing-Union, whose value is always equal to 0). Orange bars show the results for the Nice metric and green bars for the Clear metric.

Grouping

As said earlier, we only found significant results for the grouping value for Text content and Circular Shapes. Results reveal that Overlapping-Blending and Squashing-Blending are the deformers that better represent a grouping of multiple decals, except for text content on rectangular decals. Results reveal no significant difference between these two operators. While the fact that these two operators better represent a grouping behaviour was predictable, given that they apply a blending deformation, we discovered that this blending effect was stronger on textual decals.

Shape Preference and subjective feedback

When asked which decal shape they preferred, participants' answers varied according to the content type. Rectangular shapes with rounded borders were preferred for icons (ranked first by 16 participants) and images (13). It is interesting to note that the rectangular shape was the least preferred for icons and images (ranked last 16 and 15 times, respectively), even though we are used to displaying images on rectangular windows. Instead, the rectangular shape is preferred for text (12).

Participants commented on their shape preference choice. Concerning the overall preference for Rectangular shapes with Rounded borders (for icons and images), participants stated that the shape was more "pleasant" (P3, P22) and that "the rounded rectangle seems to allow the morphing in a more smooth fashion" (P6). Concerning the preference for rectangular shapes for textual content, participants referred to their habits and reading performance: "Text in rectangles is more familiar and natural" (p7). Participants commented on circular shapes. Those that did not like the shape mostly referred to it being space inefficient: "Circular is the least optimized" (P9). Instead, those that liked it referred to the fact that the circular shape seems to be better for deformation: "Changing the shape of a circle seems more natural than warping the rectangles" (P2).

3.7.3 Summary

To summarize, our study reveals that Overlapping-Union is the deformation operator perceived to be clearer and nicer. This result holds true for all decal shapes and content. To represent a group of decals, both deformer operators using blending seem to show better results but only for textual and circular content. Concerning the shapes, participants preferred the rectangular rounded shape for icons and images and the rectangular shape for text. These results complement previous work on the perception of non-rectangular interfaces [Serrano et al., 2017, 2016, Simon et al., 2019], which mostly focused on static content.

3.8 Study 2: Decal Constraints

To validate our constraints, we conduct a simulation-based study on different application interfaces. For each application, we test the adaptation of the layout to content occlusion or a change in the overall display shape. For each of them, we compare the effects of the use of our constraints with simple warping or no deformation at all in terms of content preservation and layout simplicity. We detail below the design of the study and the collected results.

3.8.1 Study Description

Choice of experiment

Our study consists of simulating, for each interface, multiple object occlusions at predefined positions or changes in the overall display shape, then adapting the interface using different conditions and automatically computing relevant metrics on the final interface layout. We chose to conduct such automatic testing to obtain a more systematic approach to compare the different optimization approaches (e.g. control the exact position of the physical object), thereby removing all possible biases introduced by object tracking flaws or user behaviour in a real setup.

Interfaces, constraints and conditions

We implement four different interfaces to test the effects of our constraints: an Image Viewer, a Mind Map, a 3x3 Grid of Images and a 5x5 Grid of folders (see Figure 3.16). We implement our decals with different constraint conditions. Either with only the Minimum distance constraint (namely Decals_Min) or with an additional constraint varying according to the application (namely Decals_Combined): we apply the alignment constraint on the toolbar icons of the Image Viewer and the grids of images and folders (both on the lines and columns); we apply the maximum distance constraint on groups of post-its of the Mind Map. For each interface, we also implement two baselines: No_Deformation and Warping. The No_Deformation condition corresponds to an interface that would simply be occluded without any kind of deformation or positioning behaviour. The

Warping condition is similar to those proposed in the literature [Cotting et al., 2006] and consists of deforming the whole interface.

Concerning the disruption conditions, we either emulate a physical object covering the interface or change the overall display shape. Both are implemented as part of the gamut, i.e. through a gamut constraint. We simulate two different physical objects: a circular cup and a rectangular book (slightly rotated). We define 9 predefined positions of the object on the interface, corresponding to different levels of occlusion on the right and top sides of the interface. For the change in display shape, we test 10 different shapes: four geometrical shapes (e.g. circle, octagon, hexagon and pentagon), four freeform shapes and two particular shapes: a doughnut (i.e. with an inner hole) and a 'separated' shape (i.e. composed of two separate parts).

Metrics and collected data

We automatically compute two metrics: content preservation and layout simplicity. The content preservation is calculated as the average ratio between the GUI elements' size after deformation and their initial size. For layout simplicity, we use the formula provided by [Ngo et al., 2003]: simplicity is a function of the number of vertical and horizontal alignment points. We compute the difference between the initial simplicity and the value after deformation. Note that the Simplicity metric is not directly optimized by the solver. Hence our constraints and metrics are independent. We do not compute the simplicity for the Mind Maps application, given that the post-its are not aligned, nor for the No_Deformation condition, as it does not change the layout of the interface. In total, we generated 448 interfaces: we computed the content preservation metric for all 448 interfaces and the layout simplicity for 252 interfaces. For each one of the generated conditions, we also captured a screenshot of the interface illustrating our results.

Data analysis

We analyze the results with a Shapiro-Wilk test to determine the normality of the collected data. For all the reported data, the Shapiro-Wilk test reveals that data did not follow a normal distribution: we performed a Box-Cox transformation, but the data could not always be transformed to be normally distributed. Hence we conducted a Kruskall-Wallis test to verify if there is a significant difference in the layout adaptation technique and a pairwise Wilcoxon test with Bonferroni correction to calculate pairwise comparisons between adaptation techniques.

3.8.2 *Results*

Content preservation

The Kruskall-Wallis test reveals a significant effect of the Layout adaptation technique ($p<.001$) on content preservation. The pairwise Wilcoxon test shows that Warping and No_Deformation are significantly different than both the Decals_Min ($p<.001$ for both) and Decals_Combined ($p<.001$ for both) conditions. These results hold true for conditions involving a layout adaptation to object occlusion and a non-rectangular display area. They indicate that the use of constraints significantly improves content preservation, as illustrated in Figure 3.16.

When adapting the layout to an object occlusion, content preservation is, on average, 96% for the Decal conditions, 89% for Warping, and 89% for No_Deformation. When adapting the interface to a non-rectangular area, this difference is even more pronounced in favour of decals: overall decals provide content preservation of 95%, while Warping and No_deformation only 76%. This is partly due to the tested physical objects that only occlude a small part of the interface, which shows that the advantages of our approach are more evident as more content is occluded.

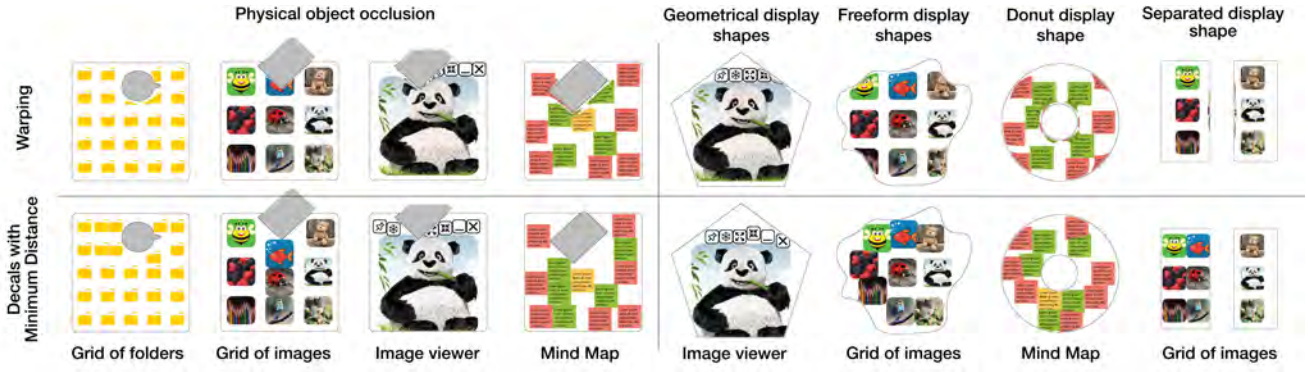


Figure 3.16: Study results reveal that, compared to regular warping (top interfaces), our decals approach (bottom interfaces) improves content preservation when adapting the layout to an object occlusion or a non-rectangular display shape.

Layout simplicity

Concerning the simplicity metric, the Kruskall Wallis test reveals a significant effect of the Layout adaptation technique ($p<.001$). The pairwise Wilcoxon test shows that Decals_Combined is significantly different than both Decals_Min ($p<.001$) and Warping ($p<.001$) conditions in terms of layout simplicity. The results did not reveal any difference between Decals_Min and Warping ($p>.05$). Again, these results hold true for conditions involving a layout adaptation to a physical object occlusion and to a non-rectangular display area. Overall the use of the Alignment constraint on Decals_Combined improves layout simplicity, as illustrated in Figure 3.17.

When occluding the interface with a physical object, the layout simplicity average is 93% for Decals_Combined (i.e. with Alignment), 84%

for Decals_Min (i.e. without Alignment), and 85% for Warping. When adapting the layout to a non-rectangular display, the simplicity with Decals_Combined is 93%, with Decals_Min 75% and with Warping 81%.

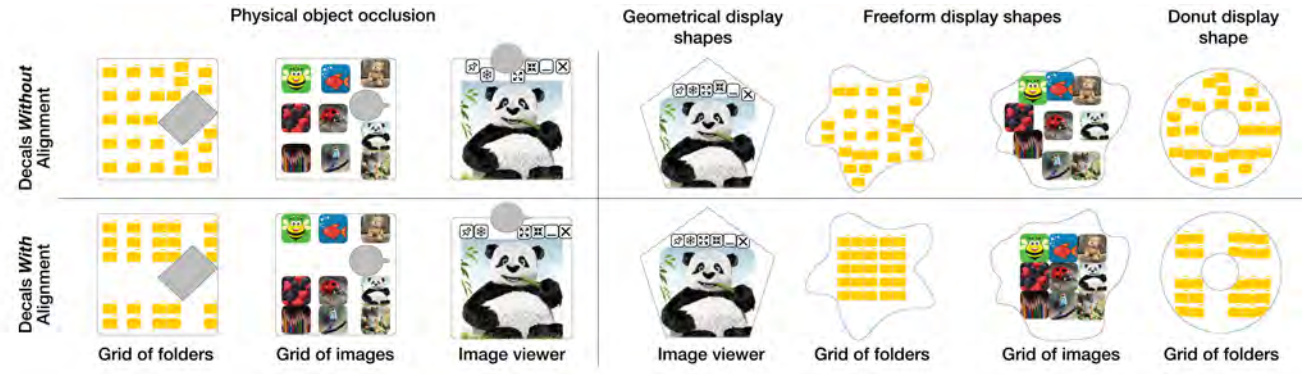


Figure 3.17: The use of an Alignment constraint (bottom interfaces) improves layout simplicity compared to using a Minimum distance constraint only (top interfaces).

3.8.3 *Summary*

Our results reveal that, overall, the use of decals preserves the interface content and the layout simplicity compared to not deforming the interface or warping the entire interface. These results hold true when the interface is occluded by an object or when the interface adapts to a non-rectangular display area. Our results also allow us to compare the performance of our solution when using different constraints and validate that using the Alignment constraint improves layout simplicity compared to using only the Minimum distance constraint.

3.9 *Study 3: User Evaluation*

We conducted a final study with the goal of collecting feedback from users regarding aesthetics, content visibility and usability of our system in a real physical setup.

3.9.1 *Study Description*

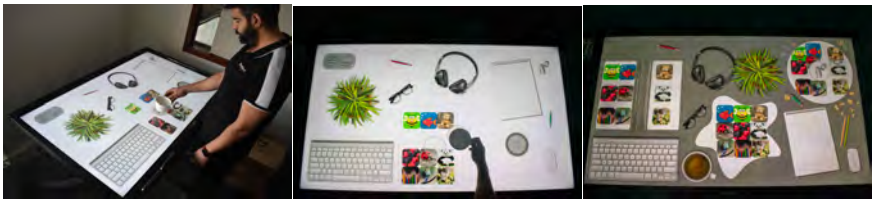
Study task and instructions

The user study consisted of two parts corresponding to the two disruption conditions: object occlusion and change in the overall display shape. In the first part (Figure 3.18-a and -b), we asked participants to grasp a physical object (a mug in our study), displace it over two predefined positions on the interface to occlude it and rate the interface on three different metrics: interface aesthetics, content visibility, and content alignment or grouping (depending on the interface). In the second part (Figure 3.18-c), the participants had to evaluate, using the same metrics, the interfaces presented in three non-rectangular display areas: a freeform shape, a

doughnut (with inner hole) and a 'separated' shape (composed of two parts).

Conditions

For the first part, we tested the same four interfaces as in our previous study, namely the Image Viewer, the 3x3 Grid of Images, the 5x5 Grid of Folders and the Mind Map. We compared two baselines (No_Deformation and Warping) against two Decals conditions (Decals_Min and Decals_Combined). As in the previous study, the Decals_Combined consisted of applying the Alignment constraint for all interfaces except the Mind Map, for which we applied a Maximum Distance constraint (which had the effect of grouping post-its of the same colour). For the second part of the experiment, we evaluated three interfaces (Image Viewer, 3x3 Grid of Images and 5x5 Grid of Folders). We did not include the Mind Maps as we posited that changing the display shape would have minimal impact on the Maximum distance constraint. We presented each interface on three freeform display areas positioned next to each other (Figure 3.18-c).



(a) User Evaluation Part 1

(b) User Evaluation Part 1 - top view

(c) User Evaluation Part 2 - top view

Figure 3.18: Two parts of the study with the effect of Alignment on 3x3 Grid of Images. (a) and (b) Physical Object Occlusion. (c) Static Freeform Display Areas

Setup

We chose to experiment using a tabletop instead of a projection-based system to remove any bias due to inefficient tracking of the physical object. We used a 65 inches multitouch tabletop (1920x1080px) at a height of 80 cm, allowing a comfortable grasp of the physical object when standing in front of it. The mug (5cm radius, 6cm height) had three soft pads (1.5cm height) attached to its bottom to be detected as a touch pattern on the touchscreen. To make participants more immersed in the environment, we imitated the presence of other physical objects around our interfaces. Moreover, the experiment was in a dark room to eliminate outside reflected lights that may interfere with object detection. Our system was running on a desktop PC (i7-4790K CPU @ 4.00GHz; 32GB; GeForce GTX 980).

Participants

Twelve participants (2 females, 10 males) took part in this study (average age of 27.75, SD = 3.2). Ten of them were local university students (1 Master's Degree and 9 PhD students), and 2 were researchers (in physics and computer science). All the participants had very low or zero experience with tabletop systems.

Study Design

The study followed a 4x4x2 within-subjects design with Interface (Mind Map, Image Viewer, 3x3 Grid of Images and 5x5 Grid of Folders), Display Condition (No_Deformation, Warping, Decals_Min and Decals_Combined) and Type of disruption (Physical object occlusion, Freeform display area) as factors. As said earlier, we did not evaluate the Mind Map interface on the freeform display shape. We counterbalanced the Interfaces and Display Conditions across participants. In total, we had 12 trials per Type of disruption.

Data collection and analysis

After each condition, participants used a 5-point Likert scale to rank their preferences on the metrics mentioned above (nice, visible content, alignment and grouping). At the end of the study, we measured the usability of each display condition with a System Usability Scale (SUS). To analyze the Likert scale results, we treated data as nominal non-parametric variables. We used a Kruskal-Wallis test to obtain the degree to which one group has higher ranks than the others. After that, we made a posthoc analysis using a pairwise Mann-Whitney-Wilcoxon test [de Winter and Dodou, 2010].

3.9.2 *Results*

For both parts of the study, the Kruskal-Wallis test showed a significant difference between display conditions ($p<.001$). In this section, for each metric and condition, we will report the percentage of participants that gave a positive score on the 5-point Likert scale, i.e. that answered *Agree* or *Strongly agree*.

Physical Object Occlusion

Figure 3.19-left illustrates the results of the Likert scale for the first part of the study, where participants moved a physical object over the interfaces. For the Nice and Visible metrics, the Wilcoxon pairwise comparison shows that there is a significant difference between the Decals conditions and the baseline conditions ($p < .001$ for all). Decals_Min and Decals_Combined were evaluated as nicer and more visible compared to No_Deformation and Warping techniques. The percentage of positive scores for the Nice metric was respectively 71% for Decals_Min, 85% for Decals_Combined, 25% for No_Deformation and 25% for Warping. The percentage of positive scores for the Visible metric was respectively 79% for Decals_Min, 88% for Decals_Combined, 6% for No_Deformation and 10% for Warping.

Concerning the content Alignment, results only revealed a significant difference between Decals_Min and the other conditions ($p < .001$ for all). All conditions were perceived to offer a better content alignment (Decals_Combined: 89%, No_Deformation: 83%, and Warping: 89%) than Decals_Min (30%).

Results were similar regarding the Grouping metric for the MindMaps application, where we only found a significant difference between Decals_Min and the other conditions ($p < .001$ for all). All conditions were perceived to offer a better content grouping (Decals_Combined: 100%, No_Deformation: 92%, and Warping: 84%) than Decals_Min (25%).

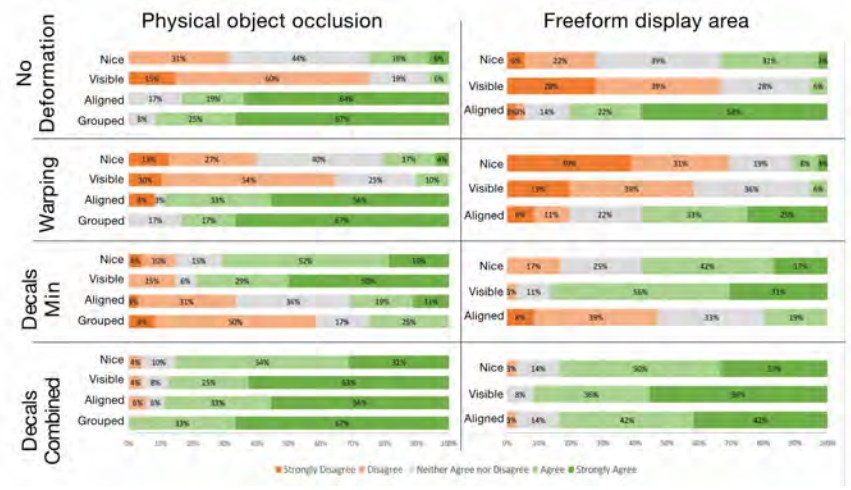


Figure 3.19: Likert scale results for the two parts of the study: Physical object occlusion (left) and Static freeform display area (right).

Static Freeform Display Areas

Figure 3.19-right illustrates the results of the Likert scale for the second part of the study, where interfaces were presented in freeform display areas. For the Nice metric, the Wilcoxon pairwise comparison reveals a significant difference between all conditions (p=0.015 between Decals_Combined and Decals_Min, $p < .001$ for all others) except between No_Deformation and Decals_Min. Overall, Decals_Combined was the nicer condition (83%), followed by Decals_Min and No_Deformation (59% and 31% respectively), with Warping being considered the less nice (11%).

For the Visible metric, the Wilcoxon pairwise comparison reveals a significant difference between all conditions (p=0.048 between Decals_Combined and Decals_Min, $p < .001$ for all others) except between No_Deformation and Warping. Decals_Combined and Decals_Min were considered to offer better content visibility (92% and 87% respectively) than No_Deformation and Warping (6% each).

Finally, regarding the content alignment, the Wilcoxon pairwise comparison reveals a significant difference between all conditions ($p < .05$ for all) except between No_Deformation and Decals_Combined. Overall, the content alignment was thought to be better when applying No_Deformation and Decals_Combined (80% and 84% respectively) than with Decals_Min (19%) or Warping (58%).

System Usability Scale

Regarding the SUS results, No_Deformation had an average score of 78.1, which corresponds to a system between "Good" and "Excellent", Warping had a score of 60.0, which equals "OK" and Decals_Min had a score of 71.9, which is "Good". Participants evaluated the Decals_Combined separately using the Alignment constraint and the one using the Maximum distance constraint (only on the Mind Maps interface): the Decals_Combined with alignment scored 80.2, whereas Decals_Combined with maximum distance constraint had a score of 84.2, which are both "Excellent".

Qualitative Feedback

The qualitative feedback for each technique helps us to understand the previous results better. Regarding the two baseline conditions, participants preferred No_Deformation over Warping because they felt the interface was more natural with no additional features, even though the content was occluded by the cup. Warping was considered as cool but not useful or usable. Regarding Decals_Min, participants mentioned that it was hard to track the distribution of the UI elements, especially for the 5x5 Grid of Folders. Considering the Decals_Combined, they noted that it was easy to predict the behaviour of the UI elements.

3.9.3 Summary

To sum up, our user study confirms that displaying interfaces using our approach offers several benefits compared to the two baselines. With our approach, the interface content is nicer and more visible than the baselines, and Decals_Combined preserves content alignment and grouping of elements. Overall, Decals_Combined is the only technique with very positive scores across all the metrics at the same time. This is valid both when occluding the interface with a physical object, as well as when adapting the interface to a freeform display area.

3.10 Discussion

In this section, we first recall the results of our three studies. Then we show an illustrative application of how our decals may be used in various scenarios.

3.10.1 Summary of study results

Our first study explored the user's perception of the decals deformations according to the decal shape, deformer and content. Our study revealed that Overlapping-Union is the deformation operator perceived to be clearer and nicer, which is an interesting result as it contrasts with the warping approach employed in previous systems. Our results reveal that the dynamic shape deformation has an impact on shape preference, leading the square

with rounded borders shape to be preferred for icons and images. These results complement previous work on the perception of non-rectangular interfaces [Serrano et al., 2017, 2016, Simon et al., 2019], which mostly focused on static content.

Our second study validated the effects of applying constraints on content visibility and layout simplicity. We showed that not only is our solution effective when a physical object covers the interface, but it allows us to adapt the overall layout to multiple display shapes. Some of these shapes are particularly challenging from a layout perspective, such as the 'doughnut' and the 'separated' shapes, and show that our solution can be easily generalized to any freeform display shape.

In our third study, we evaluated content adaptation to the change in interface shape and the presence of physical objects. According to the participants, interface content is nicer and more visible when using our approach than baselines. Moreover, preserving content alignment and grouping graphical elements were appreciated and ranked as highly positive.

Overall, our studies validated that adopting an optimization approach to place GUI elements, combined with graphical deformation, helps solve the issues of content occlusion and the non-rectilinearity of the display on pervasive interfaces. Our approach improves content preservation and layout simplicity compared to warping the whole interface, which is the approach proposed in previous works [Cotting et al., 2006].

Figure 3.20: Six illustrative applications for the Dynamic Decals.

3.10.2 *Illustrative applications*

Our approach can have many applications beyond our initial augmented classroom user scenario. We showcase in Figure 3.20 different scenarios where we applied Dynamic Decals: from left to right and top to bottom, casual image gallery on a freeform coffee table, magic desk application [Bi et al., 2011], wall projection combining virtual post-its with physical documents, floor projection of images around users in a museum exhibit,

personal augmented workspace and augmented restaurant dishware.

In addition, we created an interactive prototype to present our Dynamic Decals to the public at the event "Nuit Europeenne des chercheur.e.s 2023" (Figure 3.21). To facilitate the understanding of our research, we presented our decals as characters from Marvel Comics. We presented the constraints as "Infinity stones", where users could enable one each separately.

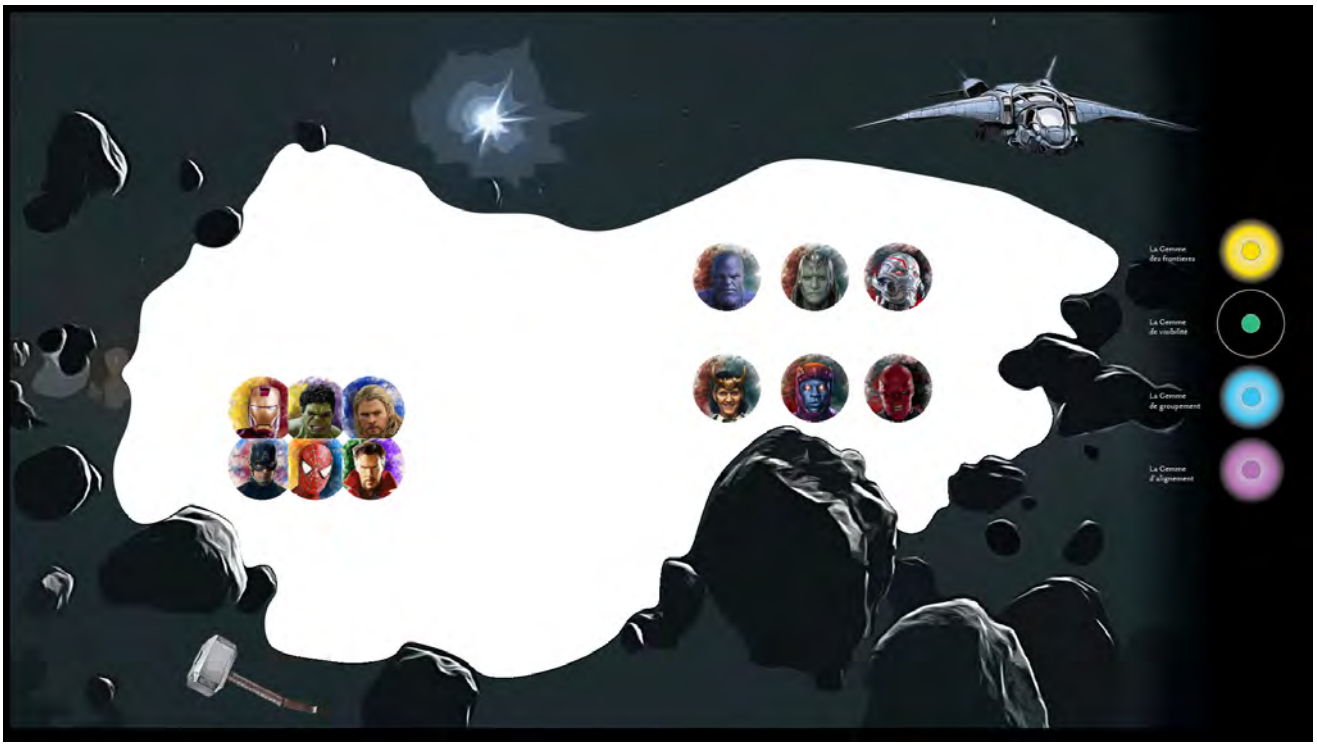


Figure 3.21: Gamification of Dynamic Decals. Three constraints are enabled: gamut, maximum distance and alignment.

3.10.3 Performances of the layout solver

We designed our prototype using a general-purpose constraint-based solver [Mellado et al., 2017b]. Regarding computation times, the optimization is fast enough to enable interactive framerates; however, we believe it can be improved in several ways, for instance, by using analytical derivative computations and parallel constraint evaluations. Regarding accuracy, the LM algorithm finds local minima of the energy function, which in some cases might lead to sub-optimal results, e.g. when no valid solution can be found (over-constrained problem: display area too small, conflicting constraints), or several optimal solutions when they exist (under-constrained problem: very large screen, missing constraints). These situations might be detected when designing the interface (to adapt the constraints) and at runtime by analyzing the shape of the display area. Visual hints could also be shown to the user to highlight parts of the GUI where the solver struggles to find a solution by analysing the optimization residuals.

3.10.4 Limitations

This research is the first step in combining graphical deformations with layout constraints to produce freeform interfaces dynamically adapting to content occlusion and non-rectangular display shapes. As such, we implemented a set of fundamental constraints for freeform GUIs with layout simplicity in mind. For the layout simplicity metric, we used the formula provided by [Ngo et al., 2003], where simplicity is a function of the number of vertical and horizontal alignment points. A first limitation of our work is that this formula, which was originally designed for rectangular layouts, may less well correlate with the simplicity of a freeform interface. The question of how well layout aesthetic metrics adapt to freeform interfaces should be specifically addressed in the future.

While the SUS results are very positive, given the task of the study (with limited interaction), the evaluated usability results need to be interpreted with caution. Other constraints can be added to address other layout properties, such as balance or symmetry [Ngo et al., 2003, Galitz, 2007], or anchors to the initial item's location (e.g. minimizing displacement of decals from their original positions). Besides layout properties, constraints could address other relevant considerations, such as higher-level semantic or task requirements. With more constraints will come the question of the scalability of the optimization. We considered a maximum combination of three constraints at the same time. Finding an optimal solution to the minimization problem becomes more difficult as we add constraints, and further studies should validate this scalability question. In our studies, we focus on adapting the decal positions. The integration of decal scale and orientation, as well as the scale of parts or even the whole interface in the solver constraints, is an interesting line of research. This would increase the space of solutions and constraints for the solver. This also requires new studies and considerations that should be the topic of new dedicated investigations.

3.11 Extensions of Dynamic Decals

During my PhD, I supervised two internships within the scope of the PERFIN ANR JCJC project. Both internships used Dynamic Decals as the base for their work and extended it either to add new features or to explore new usage scenarios.

The first internship focused on developing a module for Dynamic Decals to define the display area (gamut) dynamically. Using a depth camera, we implemented three different modes to determine the boundary of a gamut:

1. *Drawing mode* is used when the user defines a gamut by manually outlining the perimeter of a desired display area.
2. *Flooding mode* is activated by clicking anywhere in the projected area. The selected pixel defines the centre of the gamut, which will then expand outward based on the differences in depth compared to the surrounding

pixels. This can be useful if the user wants to select all the available area within a particular surface, e.g. table.

3. *Depth mode* allows adjusting a desired depth relative to the maximum and minimum depths measured in the depth field of the camera. For example, a user may select a large area (table surface) as a gamut or change it to a smaller area (a book on the table) by adjusting the target depth.

The second internship explored usage scenarios to augment reading activity in interactive tabletop systems using a tablet (Figure 3.22): the tablet would display the paper, and the surrounding tabletop could be used to display additional information (e.g. related videos or references) or excerpts from the paper. Analysing how users read scientific articles and what content is extracted from them, we proposed three models to arrange the content in a working area. These models are based on constraints and allow semi-automatic content organisation:

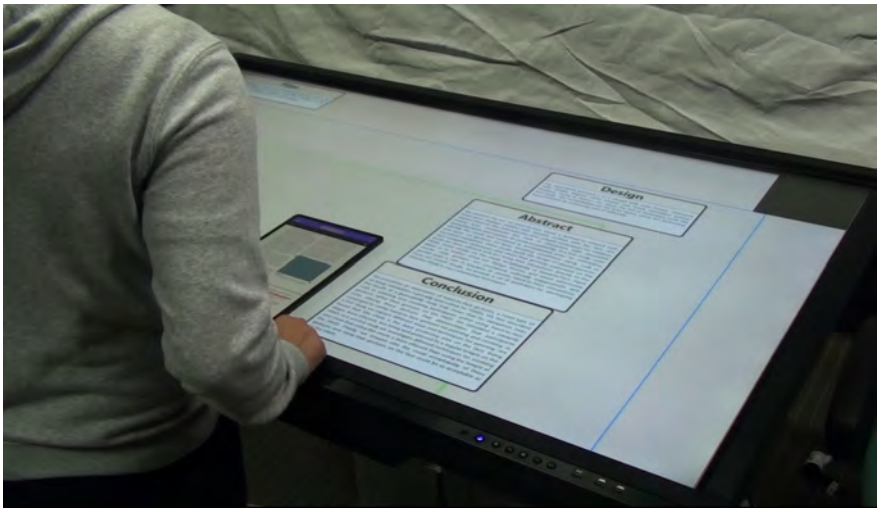


Figure 3.22: Example of using rectangle model. The abstract and Conclusion are important widgets; they are constrained within the green area that represents the most frequent access.

1. *Clock model* follows a path resembling the movement of a clock (Figure 3.23). New virtual objects representing extracted information appear next to the tablet and push older objects away along the path until they are removed. Users can freely move objects along the path or reposition them elsewhere on the screen, while an empty space on the right side allows for the fixed positioning of important information.

2. *Rectangle model* is based on the observation that users tend to position important widgets closer to them and less important ones further away (Figure 3.24). Widgets are constrained to the visible surface of the rectangles, and if they leave, they are repositioned at the closest edge. This model allows users to organize their workspace based on the importance of the content.

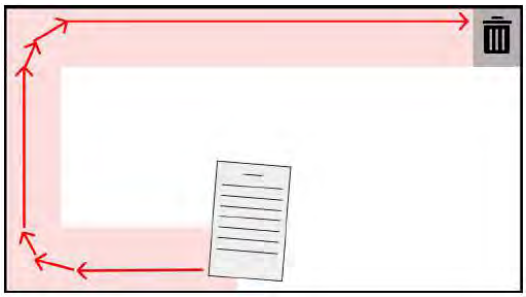


Figure 3.23: In the clock model, when new widgets appear, the older ones are gradually displaced from the tablet, following a clockwise direction along the designated path, until they are eventually removed.

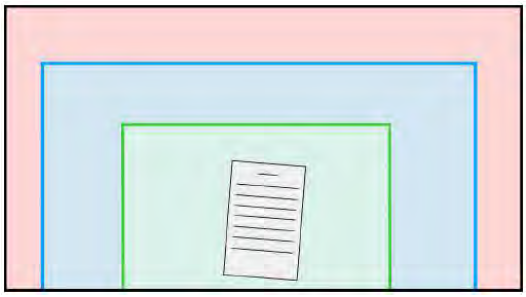


Figure 3.24: The rectangular model defines three areas: an interaction area (green) for the most frequently accessed content, a mid-area (blue) for less prioritized but still interacted content, and a storage area (red) for less important content awaiting reuse.

3. *Custom model* offers maximum user control by allowing them to create and customize their own rectangular boundaries on the screen to associate widgets with (Figure 3.25). This enables personalized workspace layouts and avoids content overflow in multi-user contexts.

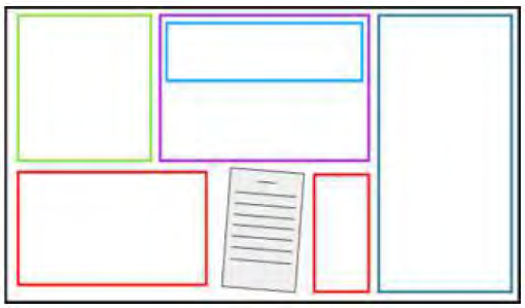


Figure 3.25: Custom model allows users to create custom layouts to allocate a place for the extracted information from the article

3.12 *Conclusion*

In this chapter, we answered the two first research questions (Table 3.1) by presenting a novel approach to tackle the challenges resulting from content occlusion and non-rectangular display areas on pervasive interfaces. Our approach is composed of two main components: (1) a dynamic layout optimisation model moving the GUI elements according to predefined constraints and user interactions and (2) a deformation model changing the shape of colliding GUI elements. We implemented the deformation model with an extension of Implicit Decals [de Groot et al., 2014], defining

different decal shapes and new reformers. An online pairwise study reveals that overall the Overlapping-Union deformer is perceived to be nicer and clearer than the others. Our optimisation model is built upon the solver by Mellado et al. [2017b] combining and minimizing constraints to update the decal positions. We validated the constraints through a study where we implemented different interfaces on which we apply content occlusion and display area changes. The results show that compared to the baselines (i.e. warping the entire interface or covering the content), our solution ensures content preservation and layout simplicity. We confirmed these results with a user study showing that, compared to the baselines, users found our system aesthetically pleasing while preserving content visibility, layout properties (alignment and grouping) and usability.

Initial research questions	Contribution
RQ1 - How to adapt the interface to freeform display areas and the presence of physical objects that can occlude the interface?	Deformation model changing the shape of colliding GUI elements
RQ2 - How to dynamically adapt the interface layout due to the presence of physical objects?	Dynamic layout optimisation model displacing GUI elements depnding on user interactions and constraints

Table 3.1: Our contribution to addressing initial research questions RQ1 and RQ2

The content of this chapter was published at a top-tier international conference in HCI, ACM ISS Conference on Interactive Surfaces and Spaces. I presented it at the conference, in Poland, in November 2021.

4

User Driven Constraints for Layout Optimisation in AR

In the previous chapter, we presented our contributions addressing the research questions RQ1 and RQ2 to optimise virtual content dynamically in projected 2D interfaces.

In this chapter, we present our efforts in addressing the two following research questions of the PhD:

- *RQ3 - How to bring interactivity into layout optimisation systems?*
- *RQ4- How to provide a set of spatial gestures to control layout optimisation while not depending on the 'legacy' of GUI?*

In this chapter, we explore the concept of user-driven constraints for immersive augmented reality layout optimisation. Our approach lets users define and set up their own constraints directly within the real-world environment. We address the previously mentioned research questions (RQ3 and RQ4) for immersive 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 [Jetter et al., 2014]. Instead, our goal is to propose a holistic, fluid [Elmqvist et al., 2011] and intuitive [Jacob et al., 2008] interaction approach allowing any user to guide the layout optimisation in the real world easily. 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.

4.1 *Usage scenario*

Before detailing our contribution, we illustrate several usage scenarios of the proposed user-driven constraints system. We envisage a use case scenario of the 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, and 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 arrange virtual objects easily but within a certain level of constraints and flexibility. John decides to use the user-driven constraints layout optimisation approach.

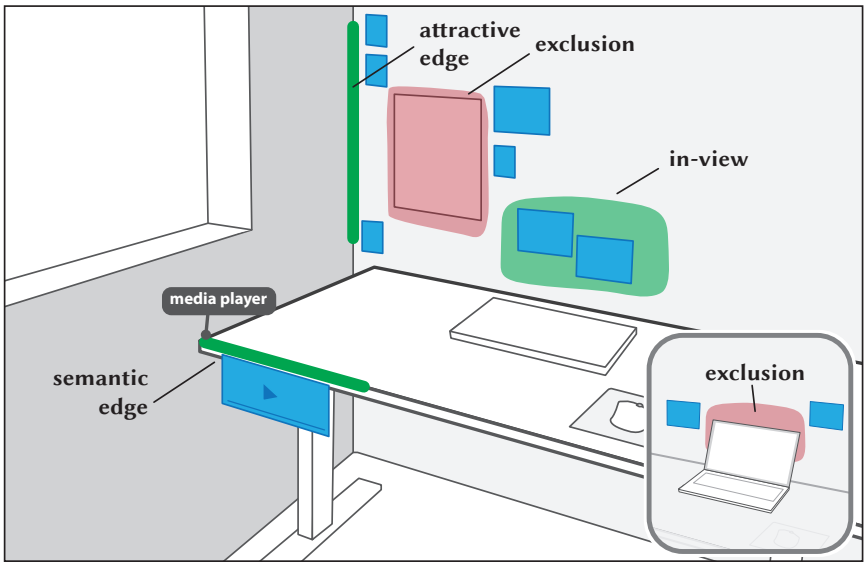


Figure 4.1: An illustration of a simple usage scenario involving attractive edge, exclusion surface, in-view surface, and semantic attractive edge. The mini figure on the bottom right illustrates the behaviour of the in-view surface when the laptop occludes the virtual content. Virtual content is illustrated in blue throughout the paper.

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 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 4.1 illustrates the final state of John's workspace.

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 4.2-a). After several new virtual objects are instantiated, i.e. new research papers that 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 4.2-b-c).

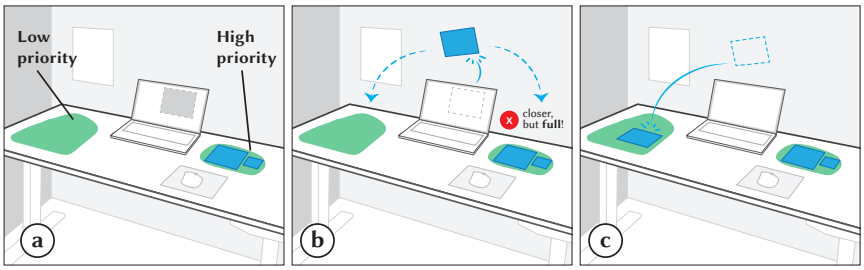


Figure 4.2: The expected behaviour of preference surfaces according to different surface priorities.

We also envisage how our user-driven constraint approach helps in collaborative scenarios, in particular, preparing a shared workspace 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 expected perspective and a second one for Sarah's expected perspective (Figure 4.3-a-b). The intersection between the two surfaces created by these frustums becomes the shared containment surface used later (Figure 4.3-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.

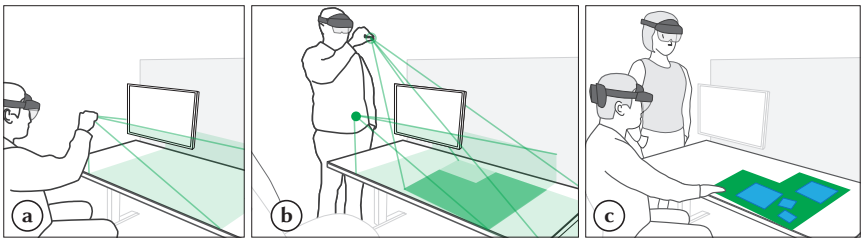


Figure 4.3: Multiple user perspectives can be authored before collaborative activities to define common containment visible to every user.

4.2 *Approach Overview and Contributions*

To answer research questions RQ3 and RQ, we first reviewed previous optimisation constraints for immersive environments in Chapter2; we selected those that can benefit from user control. Then we propose a design space for interactive immersive augmented reality layout optimisation, based on an object-action approach [Shneiderman et al., 2016] to favour fluid and intuitive immersive augmented reality interaction [Elmqvist et al., 2011, Jacob et al., 2008]. 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 an HMD. 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 users would 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.

4.3 Design Space

In this section, we describe how we derive the presented constraints in Chapter 2 into a design framework for user-driven interaction. Our design space considers three dimensions: the regions of interest, the user-driven constraints 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.

4.3.1 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. To overcome this, Liu et al. [2020b] proposed a set of design recommendations for interactive optimisation based on personal preferences or unforeseen situations. For example, users should be able to manipulate solutions directly to obtain immediate feedback. Moreover, feedback should be understandable at each optimisation stage. Following such recommendations, users can bring new knowledge into the placement approach, which brings trust and confidence in the final solution [Liu et al., 2020b].

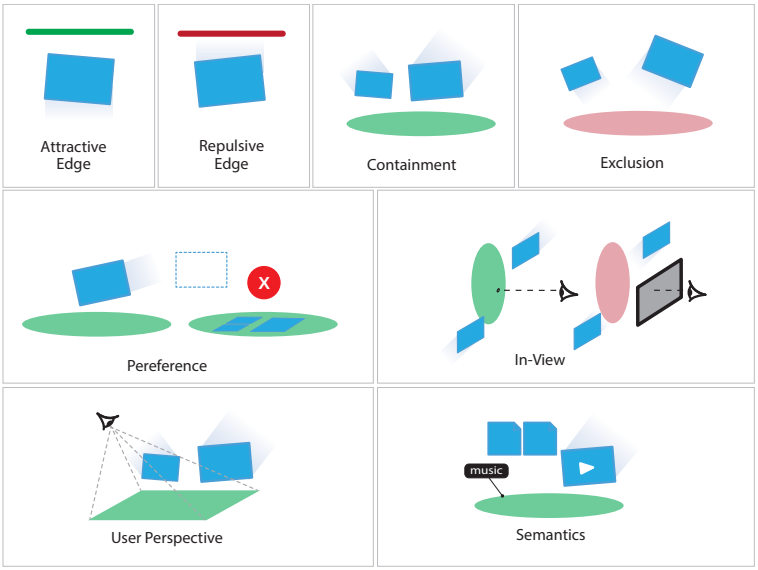
- Holistic design: our goal is to unify the interaction experience by considering all the interactions of the system and the constraints at once [Serrano et al., 2015]. This leads to a seamless and intuitive user engagement with data through direct manipulation and embodied interaction - fluid interactions [Elmqvist et al., 2011]. Also, the goal is to minimize the presence of gesture delimiters, which separate the beginning and the end of 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 Chapter 2, hand gestures are highly intuitive and can favour gesture discoverability and memorization [Yan et al., 2018].

- Direct item manipulation: our approach requires delimiting spatial surfaces or edges in 3D. It should thus support direct manipulation to promote a sense of control in dynamic virtual environments. Controlling spatial and temporal properties of virtual objects has already proven to be useful in AR in the context of authoring animations [Arora et al., 2019].

- 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 [Saidi et al., 2021].



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 4.4: 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.

4.3.2 *Regions of interest*

We consider four major regions of interest: 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). In this work, we only explore regions defined by 1D lines and 2D surfaces.

4.3.3 *User-driven constraints*

We revisited the constraints presented in Chapter 2 from a user perspective, leading to a set of user-driven constraints. 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 tag or classify the environment geometry manually 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 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 4.4 - Left. We also described the effect of optimisation for each constraint. Our constraints are defined as cost functions and 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.

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.

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.

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.

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.

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.

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).

User Perspective

This constraint allows planning for commonly used viewpoints or 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.

Semantics

This constraint allows the user to associate semantics with regions of interest. While some semantic information can be extracted automatically by using computer vision approaches [Cheng et al., 2021], having a user-driven constraint allows the user to define personal semantics. The 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 with a position on a wall where they previously hung a physical calendar. From a systems perspective, when a region has the same label as the content label, the associated cost decreases.

4.3.4 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 is detailed in Figure 4.4 - 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 4.4 - Right.

4.4 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 [Austin et al., 2020, Tsandilas, 2018, Vatavu and Wobbrock, 2015, Findlater et al., 2012], 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).

4.4.1 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 [Elmqvist et al., 2011].

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.

4.4.2 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. For instance, the containment constraint can only apply to a surface, whereas the semantics constraint could be applied to an edge, surface and volume.

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 4.4). We chose a total of 14 referents, including a command to remove the created constraints. While many more interesting referents 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 [Morris et al., 2014]. Other techniques to reduce legacy bias, such as Priming and Partners [Morris et al., 2014] 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's suggestions.

4.4.3 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's Degree student, and one was a Postdoctoral researcher. Six participants had prior knowledge of AR/VR systems.

4.4.4 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 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 performing 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 of hands and head during results analysis.

4.4.5 Design

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

4.4.6 Procedure and setup

The participants were seated at a desk in front of a wall. Such setup allows for gesture creation on horizontal and vertical surfaces, as well as in mid-air. For participants without prior knowledge of 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 [Rädle et al., 2015]. We used the illustrations of

Figure 4.4 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 $20 gift cards for their participation.

4.4.7 *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 (Figure 4.5). A first coder proceeded to create an initial gesture classification that was refined by another coder.

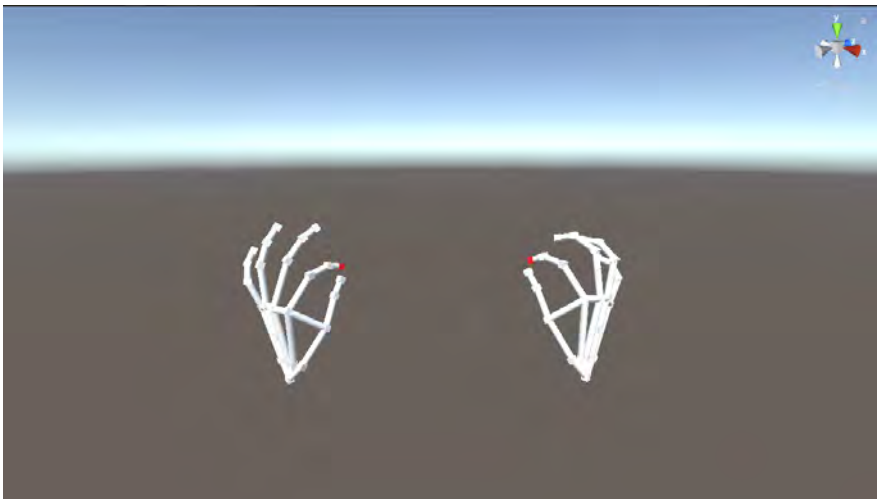


Figure 4.5: Replaying reconstructed gesture from CSV file: 2-handed pinch gesture that was proposed during gesture elicitation study to create an edge constrain

Once all gestures were coded, we measured the agreement rate (AR_i) for each referent. It was calculated using Equation 4.1 proposed by Tsandilas [2018]. Having a closer look at Tsandilas [2018] 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.

$$AR_i = \sum_{k=1}^{q} \frac{n_{ik}(n_{ik}-1)}{n_i(n_i-1)} \tag{4.1}$$

Since gesture elicitation involves subjective judgement, participants are uncertain of which gesture is the best and may respond randomly. Following Tsandilas [2018] notation, we calculate *chance agreement* (κ) using Equation 4.2.

$$\kappa = \frac{\rho_a - \rho_e}{1 - \rho_e} \text{ , where } \rho_e = \sum_{k=1}^{q} \pi_k^2 \text{ and } \pi_k = \frac{1}{m} \sum_{i=1} m \frac{n_{ik}}{n_i} \tag{4.2}$$

The output of this equation is in the range [-1;1], where values closer to 1 indicate higher chance agreement, values closer to 0 indicate low chance agreement, and values closer to -1 indicate high negative chance agreement.

4.4.8 *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.

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 in the finger used. For example, some participants started their gestures with a pinch using the thumb and index fingers, whereas others did it with the thumb and middle fingers.

Gesture agreement and preferred gestures per referent

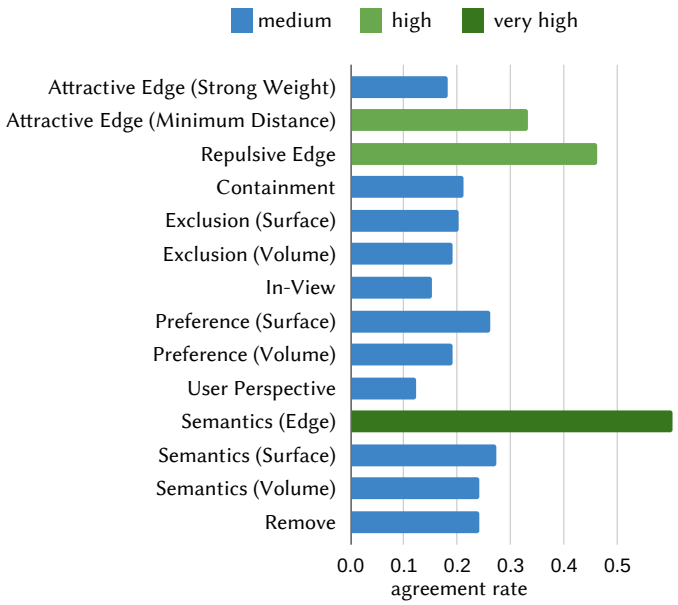


Figure 4.6: Agreement Rates for each referent. Results are colour coded according to the classification of Vatavu and Wobbrock [2015] for agreement rate values.

We computed the agreement score following the procedure of Tsandilas [2018]. The average agreement rate for all referents was 0.26. We illustrate the agreement rate for each referent in Figure 4.6. The figure is colour coded according to the classification of Vatavu and Wobbrock [2015] 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 (Figure 4.7).

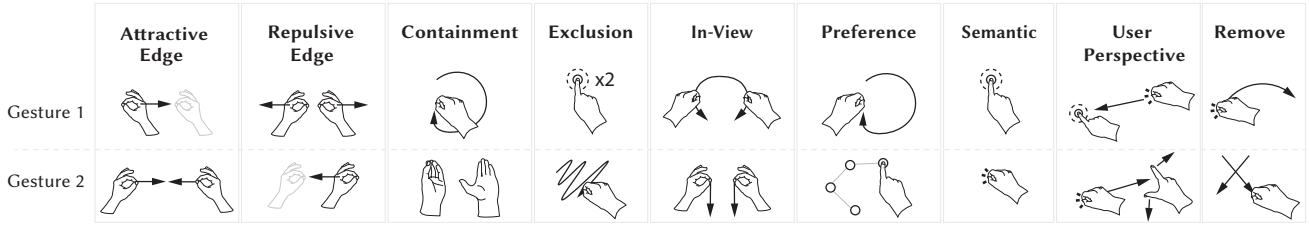


Figure 4.7: Gesture 1 and Gesture 2 are the most frequently proposed gestures for each constraint.

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.

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 the thumb and index finger after releasing a pinch.

To define an edge with a strong weight the most common technique was weight accumulation after edge creation. Before releasing a pinch, participants suggested holding and waiting, bumping pinches multiple times and doing multiple pinches. The second most used approach was to add another gesture after creating an edge. For instance, participants added a pinch or tap gesture, as well as the knock on the created edge. To emphasise the weight of the edge, some other gestures involved the movement of two hands in the same direction, palm or fist moves, and pinch with the thumb and middle finger.

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 a 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.

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.

4.4.9 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 [Shneiderman et al., 2016]: 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 immersive environments.

4.5 System Design and Implementation

According to the results, all of the participants tried to produce consistent gestures between regions and constraints. Keeping this in mind and being inspired by the proposed gestures and metaphors for constraints behaviour, 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 gestures that limits the need for delimiters and favours interaction fluidity.

4.5.1 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 4.10: 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.

4.5.2 Final set of gestures

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 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 4.8top. 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.

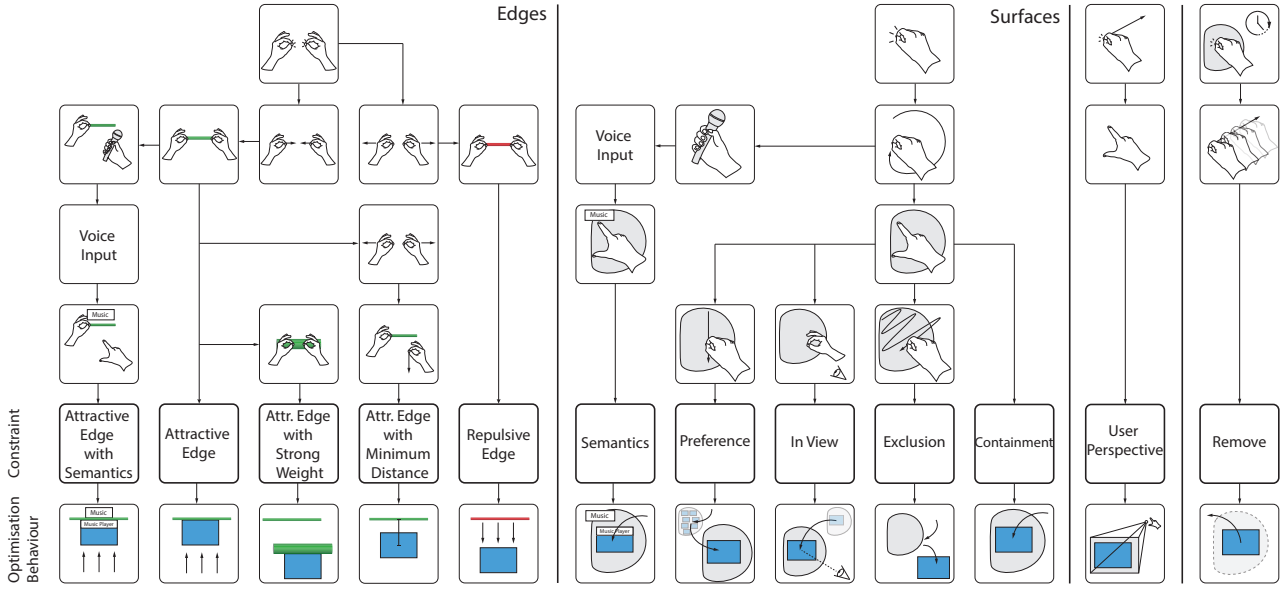


Figure 4.8: Complete diagram describing the final set of gestures of our system. All gestures begin from one of the four states shown at the top, each following a unique path to the constraints at the bottom.

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 4.8).

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 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 virtual keyboard 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.

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 centre of the view

frustum.

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 it outside of the field of view to remove it.

4.5.3 Implementation

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

Gesture

We implemented the gestures with the MRTK core services input system to detect hands and get data from pointers. Once the MRTK spatial awareness system registers the environment (Figure 4.9), we used the integrated raycast pointer to perform the 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 is 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 4.8) were implemented using the MRTK built-in capabilities.

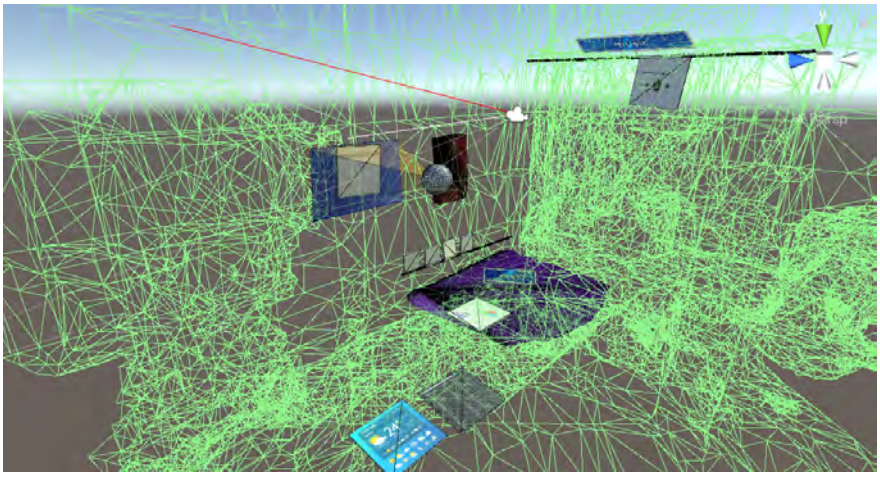


Figure 4.9: Environment mesh that is registered by Hololens 2 and constructed by MRTK

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

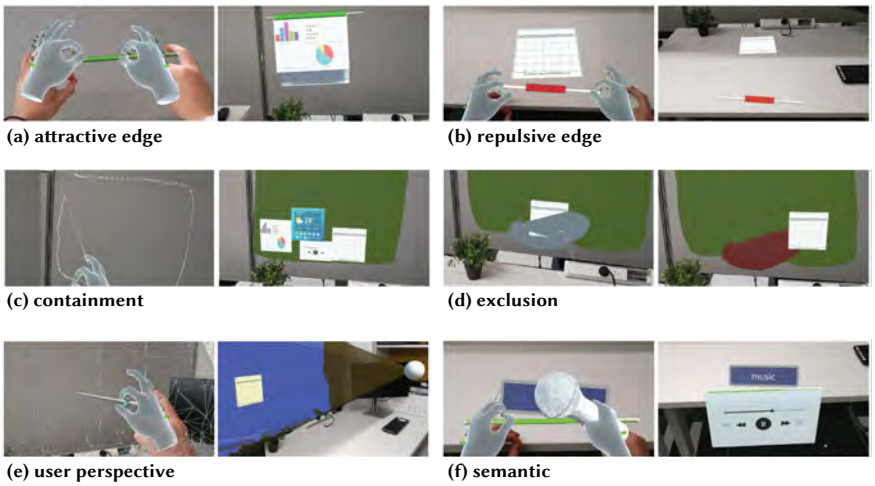


Figure 4.10: Illustration of the implementation of some of the user-defined constraints. For each constraint, we illustrate the gesture (on the left) and the resulting placement optimisation (on the right).

and final pinch positions exceeds the threshold in the direction of the inward, the edge type is considered attractive. Conversely, if the threshold is exceeded in the outwards direction, the edge type is repulsive. We also empirically introduce another threshold of 0.04m. If the distance from the 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.

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 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 to the centre of the custom Transform. In case of multiple virtual objects, they all will be attached to the centre of the container. Thus, we constrain the virtual object's movement towards the centre 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 the 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.

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 with each constraint. We adopt a distance-based approach (Figure 4.11), 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 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 redirecting content between constraints if desired.

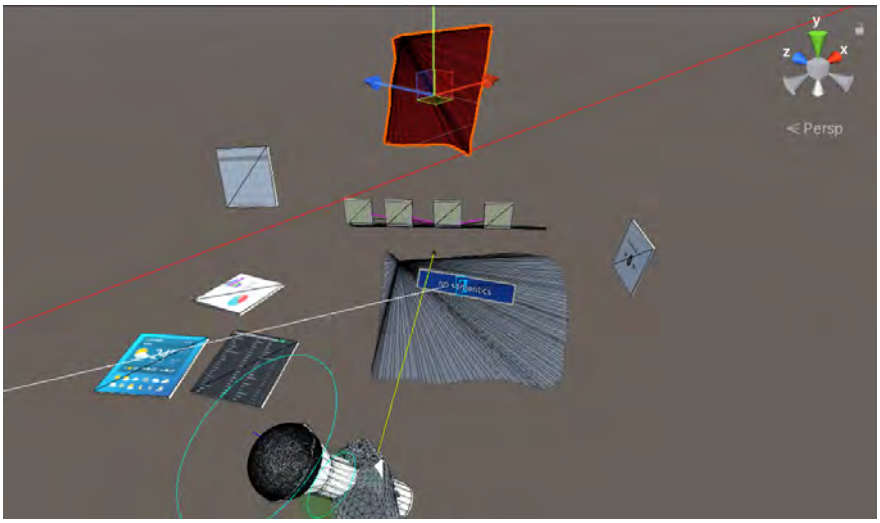


Figure 4.11: Applying a distance-based approach when combining constraints affects virtual objects only within a predefined radius. Four post-its are within the constrained distance to an Attractive Edge; pink lines illustrate the connection. The other virtual objects are too far from the constrained regions; thus, they are not under the effect of constraints. Microphone in hand implies adding semantics to the grey surface using voice input.

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.

4.6 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.

4.6.1 Study design

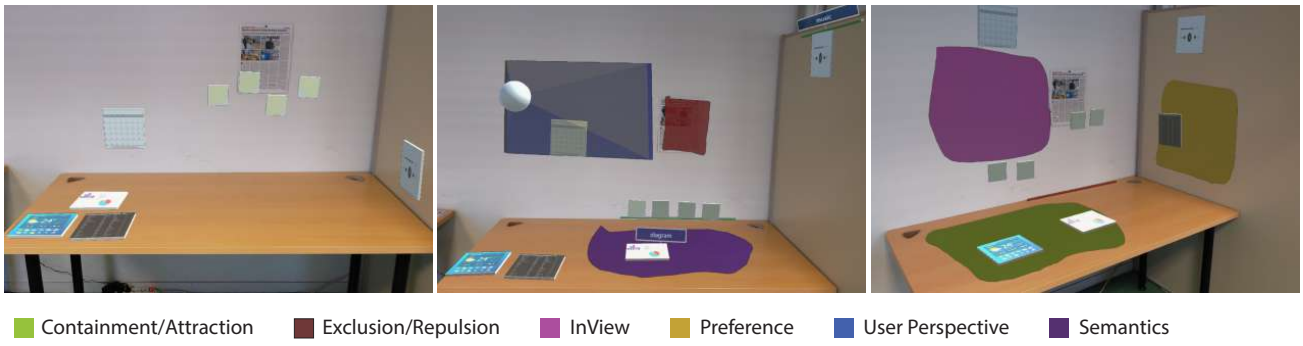


Figure 4.12: Our user study started with the virtual content manually distributed in the space during the warm-up activity (left). Then users had to arrange the content in two steps, either using a manual approach or with our user-drive constraints. With our approach, the first step (centre) involved Exclusion, Attractive Edge, Semantic Edge, Semantic Surface and User Perspective constraints; the second step (right) involved Repulsive edge, Preference, Containment and In-View constraints.

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 4.12-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, so as not to hinder the manual condition. When using the user-driven constraints, each step involved a different set of constraints (see Figure 4.12 - centre 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 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.

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 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 as 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.

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.

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's Degree students. 3 participants had prior knowledge of AR/VR systems, and none of them had prior knowledge of constraint-based optimisation.

Study Design and procedure

The study followed a within-subject design with the Technique as the 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 of 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.

Data Collection and Analysis

After each Technique, we asked participants to fill out 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 [Tapal et al., 2017, Lu and Xu, 2022]). To analyze the NASA-TLX and the Agency results, we performed t-tests.

For the user-driven constraints, we asked participants to provide 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.

4.6.2 Results

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

Gestures

Figure 4.13 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.

Figure 4.13: Likert scale results for the gesture ranking

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 a specific container for specific things (music, 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 in the perceived level of control over the content placement using

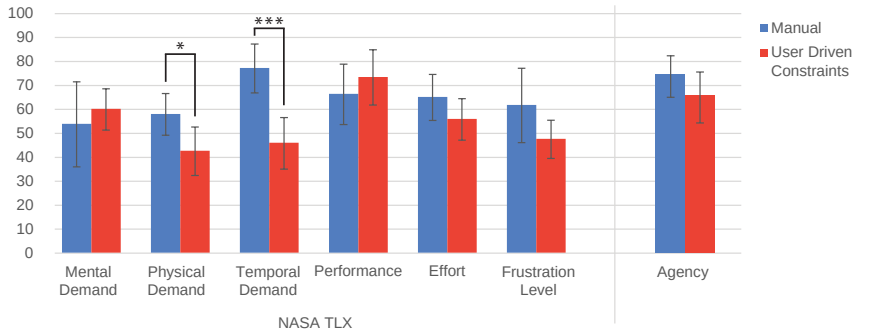


Figure 4.14: NASA TLX and Agency rankings for each technique.

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 4.14) 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 behaviour 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*".

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 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 reduced FoV and hand recognition inaccuracy. Some participants mentioned that if they performed the 2-step gestures too slowly, 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 the 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 letting the users define the direction of attraction or repulsion instead of using the surrounding surfaces.

Summary

This summative study allowed us to collect first feedback on our gestures and the interest in our approach. Overall, the gestures were appreciated and found easy to perform. Our user-driven approach was preferred to a manual arrangement and required a 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.

4.7 Discussion

4.7.1 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 the 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 a repulsive spring was created from right to left. The same is for containment and exclusion region; one is created clockwise, the other counterclockwise. Second, participants wanted the interaction to be consistent for creating regions and constraints. For instance, the semantic label was added with the same gesture for all types of regions. 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 [Ardito et al., 2014]. Our elicitation study allowed us to avoid the use of menus or GUI widgets, letting users maintain their focus on the physical surroundings. Third, another interesting observation is that some users unconsciously defined weight by mapping their fingers to prioritise the regions: index finger - high priority, pinkie finger - low priority. For example, to create a containment surface, they used pinch with thumb and index finger, whereas for exclusion surface, thumb and pinkie finger.

4.7.2 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 us to fulfil

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 in bringing interactivity into optimisation approaches (Human in the loop objective). Overall, people could use 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 [Parilusyan et al., 2022].

4.7.3 Virtual content beyond small 2D widgets

In this work, we demonstrate our approach using virtual 2D widgets such as post-it notes, weather, music or calendar widgets. Our rationale 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 of the content. 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 [Simon et al., 2019], or even to the combination with content displayed on surrounding screens [Perelman et al., 2022]. 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 the definition of volumetric constraints (instead of only surfaces). Very large windows may be difficult to fit within user-defined surfaces or conflict with others.

4.7.4 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 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.

4.7.5 *Future needs for robust and flexible layout optimisation*

The goal of our current system implementation is to demonstrate the general approach of user-defined constraints from the perspective of 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 of an external solver, such as the one developed by Mellado et al. [2017a], which has already been successfully used for optimizing the placement of 2D widgets on projected interfaces [Niyazov et al., 2021]. Future works may address the non-trivial challenge of extending this 2D optimization to 3D or predicting the impact on user performance [Cabric et al., 2021a]. 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.

4.8 *Conclusion*

In this chapter, we answered our research questions RQ3 and RQ4 (Figure 4.15). We proposed a contribution to let users define the constraints to optimize content placement in augmented reality environments. To address RQ3, we presented a design space for user-driven constraints, defining three dimensions: the region of interest, the user-driven constraints themselves, and the constraint parameters. To address RQ4, we explored this design space by conducting a user elicitation study where we asked participants to propose gestures for each user-driven constraint. Using the gesture elicitation study results, 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 in our user-driven approach and the gestures.

Initial research questions	Contribution
RQ3 - How to bring interactivity into layout optimisation systems?	Design space defining three dimensions: - user-driven constraints - region of interest - constraint parameters
RQ4 - How to provide a set of spatial gestures to control layout optimisation while not depending on the 'legacy' of GUI?	Gesture elicitation study to explore design space Implementation of a complete gesture set and the corresponding content optimisation

Figure 4.15: Our contribution to addressing initial research questions RQ3 and RQ4

The content of this chapter was published at the top-tier international

conference in HCI, ACM CHI Conference on Human Factors in Computing Systems. I presented it at the conference, in Germany, in April 2023.

5

Conclusion and Future Work

5.1 Thesis Summary

This thesis on ubiquitous interaction is at the intersection of Human-Computer Interaction and Computer Graphics. In this work, we adopted a user-centred approach to address the challenges of pervasive freeform interfaces: interface adaptation to a freeform display area, dynamic change of display boundaries, lack of user control over placement optimisation and providing spatial interaction using hand gestures. These challenges lead to the four research questions.

The first two research questions (*RQ1: How to adapt the interface to freeform display areas and the presence of physical objects that can occlude the interface? RQ2: How to dynamically adapt the interface layout due to the movement of physical objects?*) were addressed in our first work, Dynamic Decals. We presented a novel approach to decompose the interface into multiple freeform UI elements and optimise their positions dynamically using constraints. Our approach consists of two main components: a dynamic layout optimisation model that moves GUI elements based on predefined constraints and user interactions, as well as a deformation model that alters the shape of colliding GUI elements. With an expansion of de Groot et al. [2014],

we implemented the deformation model defining decal shapes and new deformers. Our online pairwise study showed that the Overlapping-Union deformer was perceived to be the best option in terms of being nice and clear. Our optimisation model was built upon the solver by Mellado et al. [2017a], which combines and minimizes constraints to update the decal positions. We validated the constraints through a study that applies different interfaces with content occlusion and display area changes. Our study results showed that our solution ensures both content preservation and layout simplicity compared to the baselines. Our user study also confirmed that users found our system aesthetically pleasing while preserving content visibility, layout properties, and usability. Overall, our approach represents a significant advancement in addressing the challenges of content occlusion and non-rectangular display areas on pervasive interfaces.

To address the other research questions (*RQ3: How to bring interactivity into layout optimisation systems? RQ4: How to provide a set of spatial gestures to control layout optimisation while not depending on the 'legacy' of GUI?*), our second work, User-driven Constraints, explored the concept of enabling users to define spatial constraints for optimizing content placement in augmented reality (AR) environments. In this work, we introduced a design space for user-driven constraints, which includes the user-driven constraints themselves, the region of interest, and the constraint parameters. To explore our design space and potential gestures for implementing it, we conducted a gesture elicitation user study. The results of the study were used to design and implement a set of gestures for content optimisation based on user-driven constraints. Finally, a controlled summative user study was conducted to verify the feasibility of our user-driven approach and the associated gestures.

5.2 Future Work

Our works open new promising directions for enriching HCI research with Computer Graphics methods and optimisation models for the design and implementation of new pervasive freeform user interfaces. In this section, we describe future research opportunities that can be pursued in the short term for each of our works, medium-term perspectives and new ideas for future long-term projects.

5.2.1 Short Term

Dynamic Decals

Our work on Dynamic Decals offers promising possibilities for creating freeform interfaces that dynamically adapt to content occlusion and freeform display shapes. We envision further improvements such as defining aesthetic metrics for freeform interfaces, implementing new constraints, further investigating the impact of distorted content and facilitating the integration of Dynamic Decals into pervasive interfaces.

Aesthetic metrics for freeform interfaces. Aesthetics or visually pleasing compositions of graphical user interfaces play a crucial role in content perception and interaction. The aesthetic metrics are well-defined for rectangular screens [Ngo et al., 2003]. However, changing the overall display shape and consequently adapting the content has been shown to have an impact on perceived layout aesthetics. There is still a lack of knowledge on how existing interface metrics can be adapted to freeform interfaces or if there will be a need to introduce new metrics. While some metrics, such as density, can directly apply to freeform displays, others, such as balance, symmetry, or simplicity, might be perceived differently on non-rectangular interfaces. Thus, we envision future research direction to define interface aesthetics by developing computational approaches that preserve aesthetic properties when changing the geometry of a display shape.

Implementation of new constraints. Another important aspect of GUI interfaces, beyond layout properties, is that GUI elements can be a part of a hierarchy. Implementing constraints on a hierarchy of objects poses a series of challenges. Still, it can also allow for a more efficient optimization by clustering the problem resolution to a subset of the decals, as already demonstrated for position-based dynamics [Müller, 2008].

Moreover, we plan to investigate other approaches to define constraints: in Dynamic Decals, we implemented position-based constraints, but we could also consider using other types of external constraints, such as user preference or virtual content placements with respect to each other.

Impact of content distortion. We also plan to investigate further the impact of presenting distorted content on Decals: while distortion has been used in HCI for a long time [Leung and Apperley, 1994], it may be better suited for certain types of content. For instance, presenting distorted text will most certainly affect its readability: the question of how to best present text on freeform surfaces has already been explored before [Serrano et al., 2016], underlining the need for proper scrolling techniques.

Integration of Dynamic Decals. Finally, a future research question would address the challenges to ease the integration of our approach into pervasive interfaces. The first step is the creation of an API to program decals and their constraints. Such a toolkit could be coupled with a projection-based approach [Raskar et al., 1999, Johnson and Fuchs, 2007, Wimmer et al., 2010] and a depth camera tracking, as implemented in an internship in the context of this PhD project.

User-driven Constraints

Our research on user-driven constraints reveals new opportunities for introducing user interaction into optimization-based approaches for spatial UI arrangement in immersive augmented reality environments, which can inspire HCI researchers interested in computational approaches.

In our work, we created a design space along three dimensions: user-driven constraints, regions of interest and constraint parameters. We only explored a part of our design space through a gesture elicitation study. According to the results of this study, participants tried to reduce the complexity of interaction by using simple gestures, often derived from or combined with other gestures. Moreover, they wanted the interaction to be consistent for creating regions and constraints. However, exploring the design space in depth and introducing more gestures to cover all constraint possibilities (combinations of regions and constraint parameters) will lead to more complex interactions. Therefore, there is a need to explore the tradeoff between the complexity of the gesture set and the level of control of constraint parameters.

Additionally, our prototype only allows users to create regions with user-driven constraints. However, it may be desirable for individuals to modify the constraints of existing regions. For example, a containment surface could be changed to an in-view surface, or an attractive edge could be transformed into a repulsive one using a simple interaction. Furthermore, it may be necessary to alter a constrained region when arranging the surrounding space. Users may want to adjust the shape or size of an existing region, modify the length of an edge, or relocate the region entirely. Thus, future research could explore hand gestures not only for creating constrained regions but also for modifying them.

Next, we envisage future research to study the balance between hard and soft constraints. In our system, we implemented only hard constraints and adopted a distance-based approach. Only virtual objects within the distance threshold are affected by constrained optimization. However, we did not investigate the combination of multiple constraints with a soft constraint approach, e.g. when the solver does not lead to an optimal solution but to a compromise between multiple constraints [Niyazov et al., 2021]. An unanswered research question is what the user expectations of such optimization are and how to allow users to change from hard constraints to soft ones if needed.

Moreover, future work may address the question of switching between optimisation modes. There are two possible options to optimise the content layout: iterative or sequential. The iterative approach separates the constraints creation from the content layout optimisation, whereas the sequential does both at the same time. In our system, we adopt the second approach, i.e. the surrounding content position is dynamically optimised as the user creates the constraints. However, it may be interesting to explore an iterative approach, which would allow the user to clearly distinguish the environment setup phase from the phase of interaction with the content. This iterative approach would need to address the challenge of how to let participants know where the content would be placed after creating several constraints.

Last, there is a future need for robust and flexible layout optimisation. Our current system implementation aims to demonstrate the general

approach of user-defined constraints from the perspective of gestural interaction. In User-driven constraints, we used a built-in MRTK solver that has some limitations, mainly 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. Given the complexity and level of details of real-world geometry, developing a more robust optimisation system will require to use of an external solver, for instance, that utilises multi-objective optimisation techniques [Todi et al., 2016, Lee et al., 2021a]. Such techniques can be based on optimization algorithms like NSGA-III [Blank et al., 2019] to find a range of Pareto [Marler and Arora, 2004] optimal solutions and be integrated with MR adaptation tools, such as AUIT [Evangelista Belo et al., 2022]. The set of optimal solutions helps to minimise stress on the user's body caused by ergonomics [Evangelista Belo et al., 2021] and maximize the relevance of context-dependent information [Lindlbauer et al., 2019]. Applying such multi-objective optimisation lets users choose the preferred solution; however, it opens a new question of how many solutions to present and how it is correlated with the cognitive load.

5.2.2 *Medium Term*

From a medium-term perspective, our two previous contributions to pervasive freeform user interfaces could be combined. In Dynamic Decals, we focused on content organisation within one display area, whereas in User-driven Constraints, we focused on the content arrangement between multiple display areas.

This combination could lead to the question of how to arrange content within one display and between multiple display areas all at once. Such a blend would benefit from a holistic and natural interaction allowing users to easily guide the layout optimisation in the real world while preserving aesthetics and keeping the content within every region well-organised. However, creating such a complex system by introducing additional features requires a device with robust hand gesture recognition. Implementing new gestures to control various parameters in our current prototype would be challenging, as we relied on the native Hololens 2 gestures. We believe further technological advancements, such as Apple Vision Pro, can overcome this limitation. By recognising a wide range of hand gestures and also detecting them outside the field of view, individuals could control multiple parameters more intrinsically to arrange the virtual content in immersive environments. For instance, when considering content alignment and grouping within regions (edges or surfaces), there is a question of how to naturally apply content organisation without relying on explicit gesture delimiters. Also, further research must address the challenges of applying content organisation to multiple regions. This entails exploring when and how to simultaneously apply such content organisation to multiple surfaces and edges. In addition, users might want to duplicate a content organi-

sation from one region to another, enabling efficient content arrangement across the interface. Addressing these challenges and research questions will contribute to the creation of well-organised immersive environments, which could be a crucial step in making such environments productive for real-world applications and scenarios.

5.2.3 *Long Term*

Research on pervasive freeform user interfaces has begun to receive attention in recent years, unlike traditional rectangular interface layouts, which have been designed and studied for decades. Therefore, developing freeform user interfaces is still complex; it often requires multi-disciplinary knowledge. As such, this field represents a very interesting opportunity for conducting interdisciplinary research and bringing together different communities beyond HCI.

Combination of user-driven optimisation and predictive models

A combination of user-driven optimisation and HCI predictive modelling approaches could facilitate the interaction in such pervasive environments. HCI predictive methods are based on the users' cognitive models to anticipate how users interact with computers. For example, GOMS (Goal, Operation, Method, Selection) describes cognitive processes to achieve a specific goal [Card, 2018]. Several variations of the GOMS model exist and can be applied depending on the task. For instance, KLM (Keystroke-Level Model) estimates the time to perform a particular task [Card et al., 1980]. TLM (Touch Level Model) is an updated KLM version that predicts users' task performance on touchscreens [Rice and Lartigue, 2014]. Further extension of KLM [Cabric et al., 2021b] predicts the time of pointing and mid-air interactions in immersive environments. Another predictive model is Fitts' Law, which focuses on the speed-accuracy trade-off when accessing UI elements [Fitts, 1954]. Derived from Fitts Law, Steering Law describes the time required to navigate a specified trajectory while keeping the pointer within the boundaries of UI elements [Yamanaka et al., 2017]. The main goal of these models is to minimize the time required to finish tasks by removing unnecessary or time-consuming interaction steps and ensuring that UI elements are simple to navigate.

Applying these predictive models could allow designers to predict and evaluate the efficiency and usability of different content placement strategies within immersive environments. Moreover, by analysing users' behavioural patterns, this approach can help designers arrange virtual content with user expectations and facilitate intuitive interactions within the immersive environment. However, combining these predictive models with user-driven optimisation introduces additional challenges in optimal content placement since user preference must be considered. For example, a user creates a constraint to position a virtual object on the wall at a certain height. In this case, the predictive model may suggest adjusting it a bit higher based on the analysis of user behaviour or on the cost of pointing

at certain items. This raises several challenges, such as how to allow the user to understand and visualize the outcome of the predictive models, how to combine user-defined constraints with predicted parameters, or how and when to limit the number of iterations between prediction and optimization.

Explainable Optimisation for Layout Management

Computational Approaches such as Machine Learning (ML) or constraint-based optimisation are needed to generate and adapt pervasive freeform UIs. However, ML introduces challenges in interpreting and explaining its learning process, prediction rationale, and internal workings, leading to its characterization as a "black box" [Castelvecchi, 2016]. Given the necessity of explaining the decisions made by these learning models to end-users, Explainable Artificial Intelligence (XAI) plays a crucial role. XAI utilizes tools, techniques, and algorithms to provide explanations for black-box models, unveiling their behaviour and underlying decision-making mechanisms [Alicioglu and Sun, 2022]. By employing visualization techniques, model and prediction explanations can be presented in a more understandable and explainable manner. However, revealing too many of these processes can become overwhelming, making it difficult to understand them. Therefore, it is important only to uncover a few of the underlying processes.

Similarly to XAI, content placement optimisation can also be understandable to end-users. Such Explainable Optimisation helps to reveal the internal functioning of optimisation, including the relationship between constraints and constrained regions. Therefore, Explainable optimisation may address the challenge of how to provide users with the appropriate visualisation of the constraint parameters, such as the weights, without visually overloading the user's environment.

In addition, when a user wishes to interact with the constrained region, the system may highlight or outline such regions and give more details about them (capacity, minimum distance). On the contrary, when the constrained region is not in use, a future improvement could reduce the brightness or opacity of the region.

Enhancing the explainability of optimization makes it more transparent and accessible. This enables users to explore the underlying processes, understand the rationale behind optimisation outcomes, and develop trust in the reliability of optimisation results.

Virtual content management in Collaborative Optimisation

Computer-supported Cooperative Work (CSCW) is the research field that helps to use technology collaboratively while having a shared goal. For instance, shared workspaces in immersive environments allow collaboratively arranging text, pictures and other virtual content to accelerate decision-making and problem-solving. In many cases, the virtual content

is constantly updated or modified simultaneously by multiple users, depending on their personal preferences. To ensure the appropriate content placement and consider multiple user perspectives, future research must address the challenges of allowing users to create collaborative spatial layouts. The simultaneous creation of spatial layouts involves various alterations in constraints creation and content organisation.

Therefore, a version control system similar to [Zhang et al., 2023] could be added to facilitate collaboration since it is important in keeping track of the shared history of users' actions, allowing them to restore the previous state of spatial layout organisation.

Furthermore, in collaborative tasks, individuals might not want to share all the virtual content with their collaborators or share their space only with particular users. Therefore, depending on a user perspective, the layout might be organised to preserve content privacy or move it to a shared area. Also, individuals might want to add notes and elaborate on their ideas in their private space before sharing them with others. Future perspectives need to explore not only when users transition between private and shared regions but also how they do it.

Moreover, users might want to collaborate between different environments, which requires the seamless interface transition between multiple interaction spaces [Billinghurst et al., 2001]. Such Transitional Interfaces [Grasset et al., 2006] require awareness of other people involved in the collaboration. It is important to have a shared knowledge of other users' identities, their location, intentions and feedback on their actions. Future research needs to address the challenges of how to maintain virtual content organisation [Ens et al., 2015] and awareness of other collaborators when transitioning between environments.

6

List of Publications

Articles in Peer-reviewed Conference Proceedings

– 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 (November 2021), 27 pages.* https://doi.org/10.1145/3488538

– Aziz Niyazov, Barrett Ens, Kadek Ananta Satriadi, Nicolas Mellado, Loic Barthe, Tim Dwyer, and Marcos Serrano. 2023. User-Driven Constraints for Layout Optimisation in Augmented Reality. *In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 35, 1–16.* https://doi.org/10.1145/3544548.3580873

Articles in Workshop

– Aziz Niyazov, Kaixing Zhao, Tao Xu, Nicolas Mellado, Loic Barthe, Marcos Serrano, "Challenges and Opportunities of Content Optimization for Freeform User Interfaces", in *CHI 2022 Workshop: Computational Approaches for Understanding, Generating, and Adapting User Interfaces*, Apr. 2022.

– Aziz Niyazov, Barrett Ens, Kadek Ananta Satriadi, Nicolas Mellado, Loic Barthe, Tim Dwyer, Marcos Serrano, "Bringing Interactivity into

Spatial Content Layout Optimisation", in *CHI 2023 Workshop: The Future of Computational Approaches for Understanding and Adapting User Interfaces*, Apr. 2023.

7

Introduction (français)

L'interaction humain-machine (IHM) est un domaine multidisciplinaire qui se concentre sur la manière dont les humains interagissent avec les ordinateurs. Ce travail de doctorat s'inscrit dans le domaine de recherche de l'IHM et applique les méthodes de l'Infographie aux interfaces ubiquitaires, c'est-à-dire les interfaces qui peuvent être affichées à tout moment et partout. Grâce aux conseils et à l'expertise de mes superviseurs dans ces domaines de recherche, ce travail de thèse a été réalisé dans le cadre d'une collaboration entre les groupes de recherche Elipse et Storm au laboratoire IRIT, Université de Toulouse 3. Une partie de mon travail a été réalisée en collaboration avec le laboratoire DVIA de l'Université Monash à Melbourne, en Australie.

7.1 Contexte et défis généraux

L'informatique ubiquitaire correspond à la notion d'accès à l'information en tout lieu et à tout moment. L'informatique ubiquitaire ou Ubicomp est un terme créé par Mark Weiser. Dans son article intitulé "The computer for the 21st century" [Weiser, 1999], il a imaginé un avenir où les objets du quotidien seraient dotés de capacités informatiques permettant une interaction transparente par le biais de gestes naturels, des sens et de la

parole.

7.1.1 L'ère pré-ubiquité

L'histoire de l'Ubicomp a commencé avec le concept de "bureau intelligent" introduit par Vannevar Bush en 1945. Selon sa vision, les ordinateurs de bureau intelligents devraient permettre aux individus d'accéder instantanément et intuitivement aux données et de les traiter. Toutefois, les dispositifs d'entrée/sortie interactifs n'étaient pas encore disponibles dans les premières années de l'informatique ; les ordinateurs utilisaient donc fréquemment des interrupteurs et des voyants de programme pour communiquer avec leurs utilisateurs (Figure 7.1). Cela convenait aux systèmes de traitement par lots qui exécutaient un programme à la fois, le programmeur jouant souvent le rôle d'opérateur. Comme les lumières et les interrupteurs pouvaient être testés et réglés à l'aide d'une seule commande machine, les coûts d'exploitation s'en trouvaient réduits. Cependant, plus tard, une fenêtre de saisie de texte appelée console système a été ajoutée pour permettre la communication entre l'opérateur et le système. À partir des années 1960, l'interaction de l'utilisateur avec les ordinateurs s'est faite principalement à l'aide d'interfaces de ligne de commande (ILC) basées sur du texte, le principal dispositif d'entrée étant le clavier. L'apprentissage des commandes et de leur fonctionnement constituait un obstacle pour certaines personnes et n'était généralement pas très intuitif pour les utilisateurs novices. Néanmoins, certaines ILC sont encore couramment utilisées aujourd'hui, principalement par les développeurs et les administrateurs systèmes, pour lesquels la rapidité et la précision sont des facteurs cruciaux.

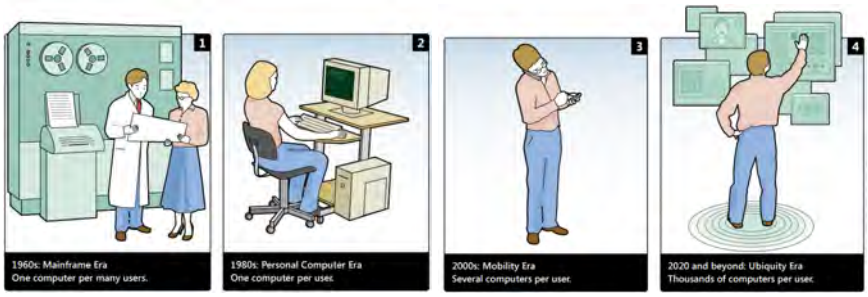


Figure 7.1: L'évolution vers l'informatique ubiquitaire : 1) 1960s : L'ère des ordinateurs centraux - un ordinateur pour de nombreux utilisateurs ; 2) Les années 1980 : L'ère de l'ordinateur personnel - un ordinateur par utilisateur ; 3) Les années 2000 : L'ère de la mobilité - plusieurs ordinateurs par utilisateur ; 4) 2020 et au-delà : L'ère de l'ubiquité - des milliers d'ordinateurs par utilisateur [Harper et al., 2008]

Avec les progrès technologiques, les nouvelles interfaces graphiques WIMP (Interfaces utilisateur graphiques basées sur Windows, Icons, Menus and Pointers) ont changé la façon dont les gens perçoivent les informations et interagissent avec les ordinateurs de bureau. Par exemple, les interfaces graphiques WIMP offrent davantage de possibilités de saisie grâce à divers dispositifs d'interaction, principalement une souris, un trackpad, un écran tactile ou une tablette graphique. Par rapport aux ILC, les interfaces graphiques WIMP sont considérées comme des interfaces plus conviviales. Shneiderman et al. [2016] considèrent que le terme "convivial" est vague et trompeur, et suggèrent d'établir des critères mesurables, tels que le temps d'apprentissage, la vitesse d'exécution des tâches ou la satisfaction subjective de l'utilisateur. Néanmoins, van Dam [1997] définit la

convivialité comme une facilité d'utilisation. En outre, van Dam [1997] décrit les autres avantages des interfaces graphiques WIMP, tels que la facilité d'apprentissage et la capacité à transférer facilement les connaissances entre les applications grâce à la cohérence de l'interface. En tant qu'interface plus intuitive et plus facile à utiliser, les interfaces graphiques WIMP facilitent l'utilisation des ordinateurs et des appareils numériques pour les tâches quotidiennes. Un autre avantage des interfaces graphiques WIMP est qu'elles sont plus efficaces pour les utilisateurs non techniques, car elles fournissent une représentation de haut niveau de l'interface, telle que des icônes ou des menus. Cependant, le développement d'interfaces graphiques bien conçues nécessite une bonne présentation de l'information, car une grande quantité de données est présentée visuellement. Sans oublier que des interfaces bien conçues peuvent améliorer la productivité, réduire les erreurs et accroître la satisfaction des utilisateurs [Chin et al., 1988, Camargo et al., 2018], tout en offrant un accès plus simple et plus rapide à l'information. Cela est d'autant plus important que l'on utilise quotidiennement de multiples interfaces.

7.1.2 *L'ère de l'ubiquité*

Par rapport aux environnements de bureau, l'informatique ubiquitaire fait référence à l'utilisation de n'importe quel appareil en n'importe quel lieu. L'informatique ubiquitaire explore les interactions multimodales et multiutilisateurs qui ne se limitent pas aux ordinateurs portables, aux téléphones mobiles, aux tablettes ou aux dispositifs portables. Par exemple, Brudy et al. [2020] a introduit SurfaceFleet pour explorer les interactions inter-appareils qui ne sont pas limitées à l'appareil, à l'application, à l'utilisateur et au temps. De plus, Houben et al. [2014] a exploré un espace centré sur l'activité des appareils multifonctionnels pour fournir un accès ubiquitaire à l'information (Figure 7.2). En outre, l'informatique ubiquitaire favorise la création d'interfaces post-WIMP [Jetter, 2013] qui améliorent les expériences et les performances des utilisateurs [Poor et al., 2016]. Le terme "post-WIMP" a été introduit par Andries van Dam et fait référence à d'autres modes de présentation de l'information et d'interaction avec l'utilisateur : *"A post-WIMP interface to me is one containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons"* - van Dam [1997]. Au lieu de cela, les interfaces post-WIMP s'appuient sur la communication en langage naturel, comme les gestes et la reconnaissance vocale, et se concentrent sur des principes de conception qui reflètent les objectifs de l'Ubicomp, comme la réduction de l'effort cognitif au cours de l'interaction.

Figure 7.2: ActivitySpace : avec un dispositif maître (ordinateur portable), les informations peuvent être déplacées entre les dispositifs ou stockées dans l'espace partagé. En outre, la configuration du système peut être préservée tout en permettant la mobilité des appareils en les épinglant dans l'environnement. [Houben et al., 2014]

De plus, le paradigme de l'informatique ubiquitaire permet d'intégrer de manière transparente les interfaces dans les objets et les environnements de la vie quotidienne, tels que les espaces publics, les habitations et les bureaux. Par exemple, dans les maisons intelligentes, les téléviseurs avec télécommande, les cadres photos, les téléphones portables, les écrans tactiles, les chaînes stéréo et les PC peuvent être utilisés de manière complémentaire. Dans le sillage de la vision ubiquitaire, Raskar et al. [1998] a introduit le concept de "bureau du futur" (Figure 7.4), dans lequel il propose de projeter des interfaces partout dans l'espace de travail. Selon cette vision, tout peut être une surface d'affichage dans l'environnement réel. Ce concept vise à créer un environnement adaptatif et personnalisé pour chaque utilisateur.

Les termes "ubiquitaire" et "interfaces pervasives" sont souvent utilisés de manière interchangeable. Ce concept a récemment gagné en popularité avec un besoin accru d'accès aux données en raison d'un contenu interactif dynamique et d'un vocabulaire d'interaction cohérent entre plusieurs dispositifs, par exemple en utilisant des gestes tactiles dans les affichages publics ou les interfaces des voitures. La notion d'interfaces ou d'affichages ubiquitaire couvre de nombreux cas d'utilisation, tels que les bâtiments intelligents, les auditoriums d'université, les musées, les salles d'exposition et les devantures de magasins. Par exemple, pour faciliter les présentations pendant les réunions, le brainstorming ou l'éducation et améliorer l'expérience interactive, Cotting et al. [2006], Jones et al. [2013] a introduit des interfaces sensibles à l'environnement basées sur la projection. En outre, les projections pourraient être utilisées pour des dispositifs interactifs mobiles afin de concevoir et d'explorer de nouveaux facteurs de forme [Brockmeyer et al., 2013]. De plus, les interfaces pervasives pourraient être utilisées dans le contexte de la maintenance pour aider les techniciens (maintenance, entretien-réparation), y compris la lecture d'incidents à distance et les simulations pour les futurs bâtiments. Le concept des interfaces pervasives peut également avoir un impact sur des millions d'utilisateurs, car elles peuvent présenter diverses visualisations de données et informations dans de nombreux domaines, tels que l'éducation, le divertissement, la santé ou la criminalistique.

7.1.3 Interfaces pervasives à forme libre

Les interfaces pervasives ne se limitent pas aux zones d'affichage rectangulaires traditionnelles. Elles utilisent des formes libres non conventionnelles, telles que circulaires ou triangulaires, et ont été explorées dans divers domaines. Ces interfaces de forme libre sont couramment utilisées dans les smartwatches circulaires commerciales et permettent de nouvelles méthodes de saisie, telles que l'interaction avec la lunette de la montre. L'iPhone 14 est un autre exemple commercial d'interface à forme libre. Dans ces smartphones, Apple a introduit un îlot dynamique, une zone de l'écran dont la forme et la taille peuvent changer pour afficher des alertes et des notifications. En outre, les interfaces de forme libre sont utilisées dans les tableaux de bord des voitures, car elles présentent des avantages fonction-

nels pour s'adapter à un environnement particulier. D'autres scénarios peuvent impliquer l'intégration d'interfaces dans des plaques de cuisson, des miroirs et des poignées de vélo et présenter des informations dans des panneaux routiers circulaires et triangulaires (Figure 7.3). Ces interfaces de forme libre peuvent également être projetées n'importe où dans l'environnement, ce qui permet une grande polyvalence dans le choix de l'emplacement optimal pour leur projection. Dans les contextes ubiquitaires, les interfaces sont sujettes à des changements dynamiques en fonction de l'environnement. Par exemple, lorsqu'un utilisateur ajoute un objet physique, l'interface s'adapte à l'occlusion. Par conséquent, les interfaces adaptatives tenant compte de l'environnement fournissent une couverture efficace de l'espace et une visibilité du contenu virtuel [Riemann et al., 2018]. En outre, les interfaces non rectangulaires offrent davantage de possibilités créatives aux concepteurs et une expérience utilisateur attrayante pour les utilisateurs [Lu et al., 2020, Basballe and Halskov, 2010]. Les interfaces non rectangulaires permettent de nouveaux types d'applications qui n'étaient pas possibles auparavant, comme les installations interactives et les expositions dans les musées [Lu et al., 2020, Lee et al., 2019].

Figure 7.3: Quelques scénarios d'application pour les interfaces pervasives de forme libre : utilisation d'un miroir de poche pour des informations privées (à gauche), d'une plaque de cuisson pour afficher des recettes (au milieu), de panneaux routiers d'avertissement pour afficher des informations destinées au public (à droite) [Serrano et al., 2016]

Les interfaces ubiquitaires de forme libre peuvent être utilisées dans des environnements plus immersifs et offrent une toile infinie pour le contenu virtuel (Figure 7.4). Ces interfaces peuvent être totalement immergées dans une réalité virtuelle (RV) ou une réalité augmentée (RA) immersive à l'aide d'un casque (head-mounted displays ou HMD) avec une superposition d'images numériques. L'interaction dans de tels environnements immersifs diffère des interfaces 2D traditionnelles non immersives. L'organisation spatiale d'un contenu virtuel nécessite différentes modalités d'entrée. Par exemple, les technologies de RA utilisant des HMD permettent d'interagir avec le contenu virtuel par des gestes de la main. Outre l'interaction manuelle, l'emplacement du contenu virtuel peut être optimisé et organisé automatiquement. Ces optimisations automatiques du contenu virtuel ne nécessitent plus d'intervention de la part de l'utilisateur et peuvent être basées sur différentes contraintes telles que l'association sémantique, le point de vue de l'utilisateur, la géométrie de l'environnement ou la persistance du contenu dans le temps.

Figure 7.4: Exemple d'Ubicomp. Concept de bureau du futur, où les interfaces sont projetées sur les surfaces horizontales et verticales.

7.1.4 Nouveaux défis

Ce travail de thèse fait partie du projet ANR JCJC PERFIN (PERvasive Freeform INterfaces), qui se concentre sur la création d'un paradigme d'interaction pour les interfaces utilisateur qui ne sont pas présentées dans des formes d'affichage rectangulaires. Dans ce travail, nous nous intéressons principalement à deux formes d'interfaces pervasives de forme libre : les interfaces projetées et les interfaces immersives de réalité augmentée (Figure 7.5). Comme illustré précédemment, ces interfaces sont caractérisées par leur **zone d'affichage libre**, leur **contenu dynamique**, leur **interaction spatiale** et leur **mise en page complexe du contenu**, ce qui a conduit à l'utilisation d'**approches basées sur l'optimisation de la mise en page**. En tant que telles, ces interfaces posent de nouveaux défis en matière d'organisation du contenu et d'interaction, comme le décrit la Section 7.2. Pour relever ces défis, il est nécessaire de réexaminer les principes fondamentaux de l'IHM recueillis au fil des années, en ce qui concerne la présentation et l'interaction avec le contenu sur les interfaces rectangulaires. Nombre de ces principes doivent être repensés pour prendre en charge les interfaces non rectangulaires. Pour atteindre les objectifs de PERFIN, cette thèse aborde le défi général de la disposition du contenu virtuel sur des interfaces pervasives de forme libre.

Figure 7.5: Une interface projetée sur une assiette (à gauche) et une interface de réalité augmentée contrôlée par les mains (à droite) - source.

7.2 *Défis et questions de recherche*

Pour relever le principal défi de l'organisation du contenu virtuel dans les interfaces pervasives de forme libre, nous le déclinons en quatre défis concrets.

Défi 1 - Adaptation de l'interface à une surface d'affichage de forme libre. Le déploiement d'interfaces de forme libre est rendu difficile par la non-rectangularité de la surface d'affichage (par exemple, une table circulaire), ainsi que par la présence d'objets (par exemple, une tasse ou un livre sur une table) qui peuvent occulter le contenu en fonction de la perspective d'observation. Il est complexe d'adapter les interfaces actuelles pour qu'elles s'adaptent mieux à l'espace disponible et en tirent profit, car les fenêtres et les widgets des interfaces graphiques traditionnelles sont principalement rectangulaires et conçus pour des écrans rectangulaires. Ce défi conduit à la première question de recherche **(QR1) : Comment adapter l'interface aux zones d'affichage de forme libre (forme extérieure) et à la présence d'objets physiques qui peuvent occulter l'interface (trous intérieurs) ?**

Défi 2 - Modification dynamique de la zone d'affichage. Dans ces contextes ubiquitaires, l'interface doit pouvoir s'adapter dynamiquement à l'espace d'affichage disponible. Cela pose un problème supplémentaire pour la forme extérieure ou les trous intérieurs, par exemple lorsque l'utilisateur déplace une tasse sur l'interface. Ces défis conduisent à la question de recherche suivante **(QR2) : Comment adapter dynamiquement la présentation de l'interface à un changement de la forme extérieure ou des trous intérieurs ?**

Défi 3 - Manque de contrôle de l'utilisateur sur l'optimisation. Permettre aux utilisateurs de contrôler l'organisation du contenu est essentiel pour un accès efficace aux données. Cependant, la disposition manuelle du contenu virtuel dans le monde réel environnant est complexe et fastidieuse. En revanche, l'optimisation automatique de l'emplacement du contenu ne nécessite aucune entrée de la part de l'utilisateur mais ne lui permet pas de contrôler l'optimisation de l'emplacement qui en résulte, même s'il a été démontré que l'ajout d'une interaction dans les systèmes d'optimisation était bénéfique et apprécié par les utilisateurs dans d'autres contextes. Par conséquent, pour garantir le contrôle de l'utilisateur, la nécessité d'un

"humain dans la boucle" [Williams et al., 2016, Roy et al., 2019] soulève la question de recherche suivante **(QR3) : Comment introduire l'interactivité dans les systèmes d'optimisation de la mise en page ?**

Défi 4 - Interaction spatiale pour contrôler la disposition de l'interface. Les environnements immersifs requièrent différentes modalités d'interaction spatiale. Par exemple, les casques de réalité augmentée permettent aux utilisateurs d'organiser le contenu virtuel à l'aide de gestes de la main. Toutefois, ces technologies reposent sur l'héritage des composants de l'interface graphique, tels que les menus contextuels et les widgets de l'interface utilisateur. D'où la nécessité d'une interaction holistique, fluide et intuitive [Elmqvist et al., 2011] qui permette à n'importe quel utilisateur de guider rapidement l'optimisation de la disposition dans le monde réel. Cela conduit à une nouvelle question de recherche **(QR4) : Comment fournir un ensemble de gestes spatiaux pour contrôler l'optimisation de l'agencement sans dépendre de l'héritage de l'interface graphique ?**

7.3 *Contribution*

Nos contributions abordent les quatre questions de recherche précédentes séparément dans deux travaux complémentaires. Notre premier travail sur les interfaces projetées, Dynamic Decals (Figure 7.6), répond aux questions de recherche QR1 et QR2. Notre deuxième travail sur la réalité augmentée, User-driven Constraints (Figure 7.7), répond aux questions de recherche QR3 et QR4.

Figure 7.6: Dynamic Decals: Pervasive Freeform Interfaces using Constrained Defofmable Graphical Elements

Pour répondre à la question de recherche 1 (*QR1 - Comment adapter l'interface aux zones d'affichage de forme libre (forme extérieure) et à la présence d'objets physiques susceptibles d'occulter l'interface (trous intérieurs)?*), nous contribuons à la décomposition de l'interface graphique en éléments graphiques plus petits appelés "decals". Les decals peuvent être automatiquement déformés lorsqu'ills entrent en collision les un avec les autres, avec des objets physiques ou avec les limites de la fenêtre d'affichage. Cette déformation permet au contenu de l'interface de prendre une forme libre et de maximiser l'occupation de l'espace. La déformation est effectuée à l'aide de déformateurs de fonctions de champ. Nous introduisons quatre nouveaux déformateurs pour modéliser les objets de l'interface utilisateur lorsqu'ils entrent en collision, en fonction de deux comportements intérieurs (écrasement ou chevauchement) et de deux comportements frontaliers (union ou mélange). Nous implémentons également trois formes initiales différentes pour les decals : circulaire, rectangulaire et rectangulaire avec des coins arrondis. En outre, nous avons utilisé trois types de contenu pour les decals (icônes, images et texte) ; cependant, l'approche générale est valable pour n'importe quel autre contenu.

Pour répondre à la question de recherche 2 (*QR2 - Comment adapter dynamiquement la disposition de l'interface en raison de la présence d'objets physiques?*), nous proposons une approche basée sur les contraintes. Nous introduisons un nouvel ensemble de contraintes pour garantir que la disposition générale des decals respecte les propriétés de disposition adéquates. Les contraintes sont définies comme des fonctions de coût mesurant des propriétés d'interface spécifiques, par exemple, les decals doivent rester dans la zone d'affichage et ne pas se superposer à des objets physiques, et les decals doivent rester à une distance minimale donnée. Lorsque l'utilisateur interagit avec l'interface utilisateur ou lorsque la zone d'affichage est modifiée, nous optimisons l'emplacement des decals en minimisant le coût introduit par ce changement, tel que mesuré par les contraintes.

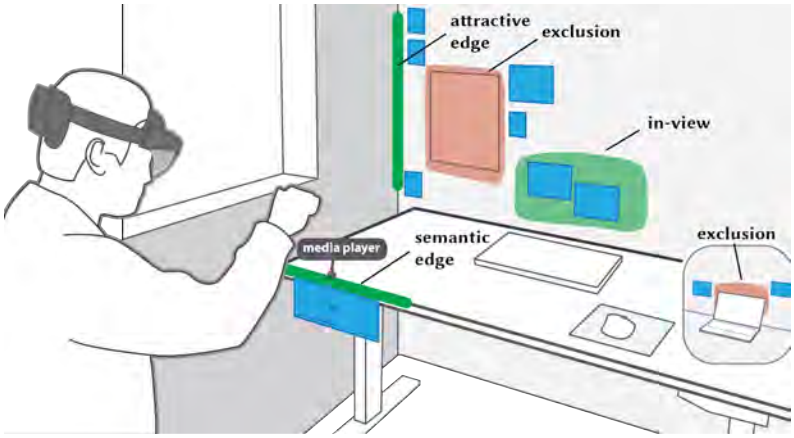


Figure 7.7: User-driven Constraints for layout optimisation in Augmented Reality

Pour explorer la question de recherche 3 (*QR3 - Comment introduire l'interactivité dans les systèmes d'optimisation de l'agencement*), nous proposons un espace de conception pour les contraintes pilotées par l'utilisateur,

c'est-à-dire des contraintes qui peuvent être définies ou paramétrées par l'utilisateur, pour affiner l'optimisation de l'agencement de la réalité augmentée de manière interactive. Notre espace de conception prend en compte les facteurs suivants : la contrainte pilotée par l'utilisateur, la région d'application de la contrainte et les paramètres de la contrainte.

Pour répondre à la question de recherche 4 (*QR4 - Comment fournir un ensemble de gestes spatiaux pour contrôler l'optimisation de la mise en page sans dépendre de l'héritage de l'interface graphique ?*), nous avons mené une étude d'élicitation des gestes pour trouver les gestes qui permettraient à l'utilisateur de définir, en une seule fois, la contrainte, ses paramètres et la région spatiale qu'elle applique. Notre approche s'éloigne des interfaces graphiques héritées qui tendent à peupler les plateformes de réalité augmentée, car elles brisent le flux d'interaction et détournent l'attention de l'utilisateur du monde réel qui l'entoure. Notre approche est également accessible aux utilisateurs ayant une expertise limitée en matière d'optimisation des contraintes.

7.4 Méthodologie et approche

Nous adoptons le processus de recherche itératif qui est largement utilisé en IHM pour concevoir, évaluer et affiner les systèmes interactifs. Bien que nous appliquions largement l'évaluation centrée sur l'utilisateur, nous utilisons également d'autres types d'approches de validation plus courantes dans l'infographie, telles que l'évaluation automatique contrôlée. L'approche itérative met l'accent sur le retour d'information et l'amélioration continus en procédant à de multiples itérations de conception basées sur les idées et les préférences de l'utilisateur. Elle permet également d'identifier et de résoudre les problèmes d'utilisabilité, en veillant à ce que le système final réponde aux besoins et aux attentes des utilisateurs.

Notre premier travail, Dynamic Decals (Figure 7.6), se situe à l'intersection de l'IHM et de l'infographie et répond aux questions de recherche QR1 et QR2. Pour concevoir, développer et évaluer nos decals dynamiques, nous avons adopté l'approche suivante :

1. Nous avons défini un nouveau type de widget déformable appelé Dynamic Decals.

2. Nous avons conçu et mis en œuvre le comportement graphique des decals dynamiques. Nous avons ensuite validé l'implémentation par le biais d'une étude utilisateur demandant aux participants d'évaluer les déformations résultant de l'application des quatre déformateurs à la forme et au contenu des decals susmentionnées en fonction de différentes mesures de visibilité et d'esthétique.

3. Nous avons conçu et mis en œuvre une approche basée sur les contraintes pour contrôler le placement des decals. Pour valider ces contraintes par le biais d'une évaluation automatique contrôlée, nous avons

mis en œuvre diverses interfaces et émulé soit un objet physique occultant son contenu, soit un changement de la forme globale de l'affichage.

4. Nous avons validé l'approche des decals dynamiques par le biais d'une étude utilisateur comparant notre approche à deux lignes de base, lorsqu'un objet physique occulte le contenu et lorsque l'interface s'adapte à une zone d'affichage de forme libre.

Notre deuxième travail sur les contraintes imposées par l'utilisateur (Figure 7.7) permet de répondre aux questions de recherche QR3 et QR4. En résumé, notre approche de développement est la suivante :

1. Nous avons créé un espace de conception pour l'optimisation interactive de l'agencement dans les environnements de réalité augmentée. Notre espace de conception prend en compte les facteurs suivants : la contrainte imposée par l'utilisateur, la région d'application de la contrainte et les paramètres de la contrainte.

2. Nous avons mené une étude d'élicitation des gestes pour explorer cet espace de conception, en nous concentrant sur les gestes que les utilisateurs effectueraient pour une telle optimisation interactive. Les participants ont proposé des gestes pour un ensemble de combinaisons de contraintes, de paramètres et de régions tout en portant un HMD.

3. Nous avons conçu un système impliquant un ensemble complémentaire de gestes pour définir les différentes contraintes. Nous avons développé notre conception dans un prototype de validation qui démontre l'application de notre espace de conception dans un scénario potentiel du monde réel.

4. Nous avons validé notre approche par le biais d'une étude sommative contrôlée, dans laquelle les participants devaient créer plusieurs contraintes en même temps pour organiser un ensemble de contenus virtuels.

7.5 Structure du manuscrit

Ce manuscrit est composé de cinq chapitres, dont le présent Chapitre d'Introduction. Dans le Chapitre 2, nous présentons une revue de la littérature de l'organisation et de l'adaptation de la présentation dans les interfaces statiques et dynamiques, des techniques d'optimisation de la présentation et de l'interaction dans les environnements immersifs. Dans le Chapitre 3, nous présentons Dynamic Decals, une nouvelle approche pour surmonter les défis des interfaces pervasives causés par une zone d'affichage de forme libre et l'occlusion d'objets. Dans le Chapitre 4, nous présentons les User-driven Constraints en abordant les défis liés à la définition des contraintes par les utilisateurs afin d'optimiser le placement du contenu virtuel dans les environnements de réalité augmentée immersifs. Enfin, dans le Chapitre 5, nous concluons ce manuscrit en résumant les contributions présentées et en discutant des perspectives de travaux futurs.

A

Bibliography

Anand Agarawala and Ravin Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 1283–1292, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933727. DOI: 10.1145/1124772.1124965. URL https://doi.org/10.1145/1124772.1124965.

Jason Alexander, Anne Roudaut, Jürgen Steimle, Kasper Hornbæk, Miguel Bruns Alonso, Sean Follmer, and Timothy Merritt. Grand challenges in shape-changing interface research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. DOI: 10.1145/3173574.3173873. URL https://doi.org/10.1145/3173574.3173873.

Gulsum Alicioglu and Bo Sun. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics*, 102:502–520, 2022. ISSN 0097-8493. DOI: https://doi.org/10.1016/j.cag.2021.09.002. URL https://www.sciencedirect.com/science/article/pii/S0097849321001886.

C. Andujar and F. Argelaguet. Friction surfaces: Scaled ray-casting manipulation for interacting with 2d guis. In *Proceedings of the 12th Eurographics Conference on Virtual Environments*, EGVE'06, page 101–108, Goslar, DEU, 2006. Eurographics Association. ISBN 3905673339.

Baptiste Angles, Marco Tarini, Brian Wyvill, Loïc Barthe, and Andrea Tagliasacchi. Sketch-based implicit blending. *ACM Trans. Graph.*, 36(6), November 2017. ISSN 0730-0301. DOI: 10.1145/3130800.3130825. URL https://doi.org/10.1145/3130800.3130825.

Carmelo Ardito, Maria Francesca Costabile, and Hans-Christian Jetter. Gestures that people can understand and use. *Journal of Visual Languages & Computing*, 25(5):572–576, 2014. ISSN 1045-926X. DOI: https://doi.org/10.1016/j.jvlc.2014.07.002. URL https://www.sciencedirect.com/science/article/pii/S1045926X14000639.

Rahul Arora, Rubaiat Habib Kazi, Danny M. Kaufman, Wilmot Li, and Karan Singh. Magicalhands: Mid-air hand gestures for animating in vr. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 463–477, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368162. DOI: 10.1145/3332165.3347942. URL https://doi.org/10.1145/3332165.3347942.

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

Greg J. Badros, Alan Borning, and Peter J. Stuckey. The cassowary linear arithmetic constraint solving algorithm. *ACM Trans. Comput.-Hum. Interact.*, 8(4):267–306, December 2001. ISSN 1073-0516. DOI: 10.1145/504704.504705. URL https://doi.org/10.1145/504704.504705.

Loïc Barthe, Véronique Gaildrat, and René Caubet. Extrusion of 1d implicit profiles: Theory and first application. *Int. J. Shape Model.*, 7: 179–198, 2001.

Loïc Barthe, N A Dodgson, M A Sabin, B Wyvill, and V Gaildrat. Two-dimensional Potential Fields for Advanced Implicit Modeling Operators. *Computer Graphics Forum*, 22(1):23 – 33, 2003. DOI: 10.1111/1467-8659.t01-1-00643. URL https://hal.archives-ouvertes.fr/hal-01538469.

Loic Barthe, Brian Wyvill, and Erwin de Groot. Controllable binary csg operators for "soft objects". *International Journal of Shape Modeling*, 10(2): 135–154, 2004.

Ditte Amund Basballe and Kim Halskov. Projections on museum exhibits: Engaging visitors in the museum setting. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, page 80–87, New York, NY, USA, 2010. Association for Computing Ma-

chinery. ISBN 9781450305020. DOI: 10.1145/1952222.1952240. URL https://doi.org/10.1145/1952222.1952240.

Jan Bender, Matthias Müller, and Miles Macklin. A survey on position based dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials*, EG '17, Goslar, DEU, 2017. Eurographics Association. DOI: 10.2312/egt.20171034. URL https://doi.org/10.2312/egt.20171034.

Adrien Bernhardt, Loic Barthe, Marie-Paule Cani, and Brian Wyvill. Implicit blending revisited. *Proc. of Eurographics, Computer Graphics Forum*, 29(2):367–376, 2010.

Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Magic desk: Bringing multi-touch surfaces into desktop work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2511–2520, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450302289. DOI: 10.1145/1978942.1979309. URL https://doi.org/10.1145/1978942.1979309.

Mark Billinghurst, Hirokazu Kato, and I. Poupyrev. The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE*, 21:6 – 8, 06 2001. DOI: 10.1109/38.920621.

Julian Blank, Kalyanmoy Deb, and Proteek Chandan Roy. Investigating the normalization procedure of nsga-iii. In Kalyanmoy Deb, Erik Goodman, Carlos A. Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed, editors, *Evolutionary Multi-Criterion Optimization*, pages 229–240, Cham, 2019. Springer International Publishing. ISBN 978-3-030-12598-1.

James F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, July 1982. ISSN 0730-0301. DOI: 10.1145/357306.357310. URL https://doi.org/10.1145/357306.357310.

Jules Bloomenthal and Brian Wyvill. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 155860233X.

Marco Blumendorf, Dirk Roscher, and Sahin Albayrak. Dynamic user interface distribution for flexible multimodal interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450304146. DOI: 10.1145/1891903.1891930. URL https://doi.org/10.1145/1891903.1891930.

Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch projector: Mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 2287–2296, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. DOI: 10.1145/1753326.1753671.

URL https://doi.org/10.1145/1753326.1753671.

Alan Borning, Richard Lin, and Kim Marriott. Constraints for the web. In *Proceedings of the Fifth ACM International Conference on Multimedia*, MULTIMEDIA '97, page 173–182, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897919912. DOI: 10.1145/266180.266361. URL https://doi.org/10.1145/266180.266361.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4): 324–345, 1952. ISSN 00063444. URL http://www.jstor.org/stable/2334029.

Susanne C. Brenner and Carsten Carstensen. *Finite Element Methods*, pages 1–47. American Cancer Society, 2017. ISBN 9781119176817. DOI: 10.1002/9781119176817.ecm2003. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119176817.ecm2003.

Eric Brockmeyer, Ivan Poupyrev, and Scott Hudson. Papillon: Designing curved display surfaces with printed optics. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 457–462, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322683. DOI: 10.1145/2501988.2502027. URL https://doi.org/10.1145/2501988.2502027.

Frederik Brudy, David Ledo, Michel Pahud, Nathalie Henry Riche, Christian Holz, Anand Waghmare, Hemant Bhaskar Surale, Marcus Peinado, Xiaokuan Zhang, Shannon Joyner, Badrish Chandramouli, Umar Farooq Minhas, Jonathan Goldstein, William Buxton, and Ken Hinckley. Surfacefleet: Exploring distributed interactions unbounded from device, application, user, and time. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 7–21, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375146. DOI: 10.1145/3379337.3415874. URL https://doi.org/10.1145/3379337.3415874.

Camille Brunel, Pierre Bénard, Gaël Guennebaud, and Pascal Barla. A time-independent deformer for elastic-rigid contacts. *Proc. ACM Comput. Graph. Interact. Tech.*, 3(1), April 2020. DOI: 10.1145/3384539.

Vannevar Bush. As we may think. *Interactions*, 3(2):35–46, mar 1996. ISSN 1072-5520. DOI: 10.1145/227181.227186. URL https://doi.org/10.1145/227181.227186.

Florent Cabric, Emmanuel Dubois, and Marcos Serrano. A predictive performance model for immersive interactions in mixed reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 202–210, 2021a. DOI: 10.1109/ISMAR52148.2021.00035.

Florent Cabric, Emmanuel Dubois, and Marcos Serrano. A predictive performance model for immersive interactions in mixed reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*,

pages 202–210, 2021b. DOI: 10.1109/ISMAR52148.2021.00035.

Murilo C. Camargo, Rodolfo M. Barros, and Vanessa T. O. Barros. Visual design checklist for graphical user interface (gui) evaluation. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 670–672, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450351911. DOI: 10.1145/3167132.3167391. URL https://doi.org/10.1145/3167132.3167391.

Marie-Paule Cani. An implicit formulation for precise contact modeling between flexible solids. In *20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1993*, pages 313–320, 1993. Published as Marie-Paule Gascuel.

Stuart K Card. *The psychology of human-computer interaction*. Crc Press, 2018.

Stuart K Card, Thomas P Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980.

Davide Castelvecchi. Can we open the black box of ai? *Nature*, 538:20–23, 10 2016. DOI: 10.1038/538020a.

Jessica Cauchard, Markus Löchtefeld, Mike Fraser, Antonio Krüger, and Sriram Subramanian. M+pspaces: Virtual workspaces in the spatially-aware mobile environment. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, page 171–180, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311052. DOI: 10.1145/2371574.2371601. URL https://doi.org/10.1145/2371574.2371601.

Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 2625–2634, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. DOI: 10.1145/1753326.1753725. URL https://doi.org/10.1145/1753326.1753725.

Dempsey Chang, Laurence Dooley, and Juhani E. Tuovinen. Gestalt theory in visual screen design: A new look at an old subject. In *Proceedings of the Seventh World Conference on Computers in Education Conference on Computers in Education: Australian Topics - Volume 8*, CRPIT '02, page 5–12, AUS, 2002. Australian Computer Society, Inc. ISBN 0909925860.

Jung-Wei Chen and Jiajie Zhang. Comparing text-based and graphic user interfaces for novice and expert users. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 11:125–9, 02 2007.

Kuan-Ta Chen, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. A crowdsourceable qoe evaluation framework for multimedia content. In

Proceedings of the 17th ACM International Conference on Multimedia, MM '09, page 491–500, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605586083. DOI: 10.1145/1631272.1631339. URL https://doi.org/10.1145/1631272.1631339.

Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, Sebastian Boring, and Saul Greenberg. Extending a mobile device's interaction space through body-centric interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, page 151–160, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311052. DOI: 10.1145/2371574.2371599. URL https://doi.org/10.1145/2371574.2371599.

Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 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, page 282–297, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386357. DOI: 10.1145/3472749.3474750. URL https://doi.org/10.1145/3472749.3474750.

John P. Chin, Virginia A. Diehl, and Kent L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, page 213–218, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0201142376. DOI: 10.1145/57167.57203. URL https://doi.org/10.1145/57167.57203.

Daniel Cotting, Markus Gross, and Markus Gross. Interactive environment-aware display bubbles. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 245–254, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. DOI: 10.1145/1166253.1166291. URL http://doi.acm.org/10.1145/1166253.1166291.

Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. Grids: Interactive layout design with integer programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. DOI: 10.1145/3313831.3376553. URL https://doi.org/10.1145/3313831.3376553.

Erwin de Groot, Brian Wyvill, Loïc Barthe, Ahmad Nasri, and Paul Lalonde. Implicit decals: Interactive editing of repetitive patterns on surfaces. *Comput. Graph. Forum*, 33(1):141–151, February 2014. ISSN 0167-7055. DOI: 10.1111/cgf.12260. URL https://doi.org/10.1111/cgf.12260.

Gerwin de Haan, Michal Koutek, and Frits H. Post. Towards intuitive exploration tools for data visualization in vr. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '02, page

105–112, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135300. DOI: 10.1145/585740.585758. URL https://doi.org/10.1145/585740.585758.

Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, page 337–340, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3540787992.

Joost de Winter and Dimitra Dodou. Five-point likert items: t test versus mann–whitney–wilcoxon. *Practical Assessment, Research and Evaluation*, 15, 01 2010.

D. A. Dondis. *A primer of visual literacy / Donis A. Dondis.* MIT Press, Cambridge, Mass. ;, mit press paperback edition. edition, 1974.

Peitong Duan, Casimir Wierzynski, and Lama Nachman. Optimizing user interface layouts via gradient descent. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. DOI: 10.1145/3313831.3376589. URL https://doi.org/10.1145/3313831.3376589.

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

Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. Ethereal planes: A design framework for 2d information space in 3d mixed reality environments. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction*, SUI '14, page 2–12, New York, NY, USA, 2014a. Association for Computing Machinery. ISBN 9781450328203. DOI: 10.1145/2659766.2659769. URL https://doi.org/10.1145/2659766.2659769.

Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. Spatial constancy of surface-embedded layouts across multiple environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, page 65–68, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337038. DOI: 10.1145/2788940.2788954. URL https://doi.org/10.1145/2788940.2788954.

Barrett M. Ens, Rory Finnegan, and Pourang P. Irani. 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*, CHI '14, page 3171–3180, New York, NY, USA, 2014b. Association for Computing Machinery. ISBN 9781450324731. DOI: 10.1145/2556288.2557058. URL https://doi.org/10.1145/2556288.2557058.

Christer Ericson. *Real-Time Collision Detection*. CRC Press, Inc., USA, 2004.

ISBN 1558607323.

João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. Xrgonomics: Facilitating the creation of ergonomic 3d interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. DOI: 10.1145/3411764.3445349. URL https://doi.org/10.1145/3411764.3445349.

João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. Auit – the adaptive user interfaces toolkit for designing xr applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393201. DOI: 10.1145/3526113.3545651. URL https://doi.org/10.1145/3526113.3545651.

Andreas Fender, David Lindlbauer, Philipp Herholz, Marc Alexa, and Jörg Müller. 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*, UIST '17, page 611–621, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349819. DOI: 10.1145/3126594.3126621. URL https://doi.org/10.1145/3126594.3126621.

Andreas Fender, Philipp Herholz, Marc Alexa, and Jörg Müller. Optispace: Automated placement of interactive 3d projection mapping content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–11, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. DOI: 10.1145/3173574.3173843. URL https://doi.org/10.1145/3173574.3173843.

Leah Findlater, Ben Lee, and Jacob Wobbrock. 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*, CHI '12, page 2679–2682, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310154. DOI: 10.1145/2207676.2208660. URL https://doi.org/10.1145/2207676.2208660.

Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

George W. Fitzmaurice, Shumin Zhai, and Mark H. Chignell. Virtual reality for palmtop computers. *ACM Trans. Inf. Syst.*, 11(3):197–218, jul 1993. ISSN 1046-8188. DOI: 10.1145/159161.159160. URL https://doi.org/10.1145/159161.159160.

Karl Flieder and Felix Mödritscher. Foundations of a pattern language based on gestalt principles. In *CHI '06 Extended Abstracts on Human*

Factors in Computing Systems, CHI EA '06, page 773–778, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932984. DOI: 10.1145/1125451.1125605. URL https://doi.org/10.1145/1125451.1125605.

Robert Fraher and James Boyd-Brent. Gestalt theory, engagement and interaction. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, page 3211–3216, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589305. DOI: 10.1145/1753846.1753960. URL https://doi.org/10.1145/1753846.1753960.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. DOI: https://doi.org/10.1002/nav.3800030109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109.

Genki Furumi, Daisuke Sakamoto, and Takeo Igarashi. Snaprail: A tabletop user interface widget for addressing occlusion by physical objects. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, page 193–196, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312097. DOI: 10.1145/2396636.2396666. URL https://doi.org/10.1145/2396636.2396666.

Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. Flare: Fast layout for augmented reality applications. 09 2014. DOI: 10.13140/2.1.3060.4167.

Wilbert O. Galitz. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. John Wiley & Sons, Inc., USA, 2007. ISBN 0470053429.

James Jerome Gibson. The ecological approach to visual perception. 1979.

Michael Gleicher. Integrating constraints and direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D '92*, pages 171–174, New York, New York, USA, 1992a. ACM Press. ISBN 0897914678. DOI: 10.1145/147156.147194. URL http://portal.acm.org/citation.cfm?doid=147156.147194.

Michael Gleicher. Briar: A constraint-based drawing program. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, page 661–662, New York, NY, USA, 1992b. Association for Computing Machinery. ISBN 0897915135. DOI: 10.1145/142750.143074. URL https://doi.org/10.1145/142750.143074.

Google. Tilt Brush. https://www.tiltbrush.com/. Accessed: 2023-03-21.

S. Gottschalk, M. C. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 171–180, New York, NY, USA, 1996. Association for Computing

Machinery. ISBN 0897917464. DOI: 10.1145/237170.237244. URL https://doi.org/10.1145/237170.237244.

Olivier Gourmel, Loic Barthe, Marie-Paule Cani, Brian Wyvill, Adrien Bernhardt, Mathias Paulin, and Herbert Grasberger. A gradient-based implicit blend. *ACM Trans. Graph.*, 32(2), April 2013. ISSN 0730-0301. DOI: 10.1145/2451236.2451238. URL https://doi.org/10.1145/2451236.2451238.

Raphael Grasset, Julian Looser, and Mark Billinghurst. Transitional interface: Concept, issues and framework. pages 231–232, 10 2006. DOI: 10.1109/ISMAR.2006.297819.

Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 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*, CHI '15, page 3933–3942, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. DOI: 10.1145/2702123.2702331. URL https://doi.org/10.1145/2702123.2702331.

Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 283–292, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307161. DOI: 10.1145/2047196.2047233. URL https://doi.org/10.1145/2047196.2047233.

Richard Harper, Tom Rodden, Y Rogers, and Abigail Sellen. *Being Human: Human-Computer Interaction in the Year 2020*. 01 2008. ISBN 0955476119.

Chris Harrison, Desney Tan, and Dan Morris. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 453–462, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. DOI: 10.1145/1753326.1753394. URL https://doi.org/10.1145/1753326.1753394.

Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. Omnitouch: Wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 441–450, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307161. DOI: 10.1145/2047196.2047255. URL https://doi.org/10.1145/2047196.2047255.

L. He and J. van den Berg. Efficient exact collision-checking of 3-d rigid body motions using linear transformations and distance computations in workspace. In *2014 IEEE International Conference on Robotics and Automation*

(ICRA), pages 2059–2064, 2014.

H. Helson. The fundamental propositions of gestalt psychology. *Psychological Review*, 40(1):13–32, 1933. DOI: 10.1037/h0074375.

Thuong N Hoang and Bruce H Thomas. Augmented viewport: An action at a distance technique for outdoor ar using distant and zoom lens cameras. In *International Symposium on Wearable Computers (ISWC) 2010*, pages 1–4. IEEE, 2010.

Kenneth E. Hoff, Andrew Zaferakis, Ming Lin, and Dinesh Manocha. Fast and simple 2d geometric proximity queries using graphics hardware. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, page 145–148, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581132921. DOI: 10.1145/364338.364383. URL https://doi.org/10.1145/364338.364383.

David Holman and Roel Vertegaal. Organic user interfaces: Designing computers in any way, shape, or form. *Commun. ACM*, 51(6):48–55, jun 2008. ISSN 0001-0782. DOI: 10.1145/1349026.1349037. URL https://doi.org/10.1145/1349026.1349037.

Kai Hormann, Bruno Lévy, and Alla Sheffer. Mesh Parameterization: Theory and Practice, 2007. URL https://hal.inria.fr/inria-00186795. This document is the support of a course given at SIGGRAPH 2007.

Hiroshi Hosobe. A modular geometric constraint solver for user interface applications. In *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, page 91, New York, New York, USA, 2001. ACM Press. ISBN 158113438X. DOI: 10.1145/502348.502362. URL http://portal.acm.org/citation.cfm?doid=502348.502362.

Steven Houben, Paolo Tell, and Jakob E. Bardram. Activityspace: Managing device ecologies in an activity-centric configuration space. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, page 119–128, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450325875. DOI: 10.1145/2669485.2669493. URL https://doi.org/10.1145/2669485.2669493.

Scott E. Hudson and Ian Smith. Ultra-lightweight constraints. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, page 147–155, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917987. DOI: 10.1145/237091.237112. URL https://doi.org/10.1145/237091.237112.

H.E. Huntley. *The Divine Proportion: A Study in Mathematical Beauty*. Dover Books on Mathematics. Dover Publications, 1970. ISBN 9780486222547. URL https://books.google.fr/books?id=rF0cj1I5KQcC.

Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-

based interaction: A framework for post-wimp interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 201–210, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580111. DOI: 10.1145/1357054.1357089. URL https://doi.org/10.1145/1357054.1357089.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and J. P. Lewis. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, SIGGRAPH '14, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329620. DOI: 10.1145/2614028.2615427. URL https://doi.org/10.1145/2614028.2615427.

Waqas Javed, KyungTae Kim, Sohaib Ghani, and Niklas Elmqvist. Evaluating physical/virtual occlusion management techniques for horizontal displays. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 391–408, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23765-2.

Hans-Christian Jetter. *Design and Implementation of Post-WIMP Interactive Spaces with the ZOIL Paradigm*. PhD thesis, Universität Konstanz, Konstanz, 2013.

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

Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. Orcsolver: An efficient solver for adaptive gui layout with or-constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450367080. DOI: 10.1145/3313831.3376610. URL https://doi.org/10.1145/3313831.3376610.

Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. Orcsolver: An efficient solver for adaptive gui layout with or-constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450367080. DOI: 10.1145/3313831.3376610. URL https://doi.org/10.1145/3313831.3376610.

P. Jiménez, F. Thomas, and C. Torras. 3d collision detection: a survey. *Computers & Graphics*, 25(2):269 – 285, 2001. ISSN 0097-8493. DOI: https://doi.org/10.1016/S0097-8493(00)00130-8. URL http://www.sciencedirect.com/science/article/pii/S0097849300001308.

Tyler Johnson and Henry Fuchs. Real-time projector tracking on complex geometry using ordinary imagery. In *2007 IEEE Conference*

on Computer Vision and Pattern Recognition, pages 1–8, 2007. DOI: 10.1109/CVPR.2007.383460.

Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 637–644, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330695. DOI: 10.1145/2642918.2647383. URL https://doi.org/10.1145/2642918.2647383.

Brett R. Jones, Rajinder Sodhi, Roy H. Campbell, Guy Garnett, and Brian P. Bailey. Build your world and play in it: Interacting with surface particles on complex objects. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 165–174, 2010. DOI: 10.1109/ISMAR.2010.5643566.

Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. Illumiroom: Peripheral projected illusions for interactive experiences. In *ACM SIGGRAPH 2013 Emerging Technologies*, SIGGRAPH '13, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323406. DOI: 10.1145/2503368.2503375. URL https://doi.org/10.1145/2503368.2503375.

Mohammadreza Khalilbeigi, Jürgen Steimle, Jan Riemann, Niloofar Dezfuli, Max Mühlhäuser, and James D. Hollan. Objectop: Occlusion awareness of physical objects on interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, page 255–264, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322713. DOI: 10.1145/2512349.2512806. URL https://doi.org/10.1145/2512349.2512806.

Janin Koch and Antti Oulasvirta. Computational layout perception using gestalt laws. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 1423–1429, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340823. DOI: 10.1145/2851581.2892537. URL https://doi.org/10.1145/2851581.2892537.

Masaaki Kurosu and Kaori Kashimura. Apparent usability vs. inherent usability: Experimental analysis on the determinants of the apparent usability. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, page 292–293, New York, NY, USA, 1995. Association for Computing Machinery. ISBN 0897917553. DOI: 10.1145/223355.223680. URL https://doi.org/10.1145/223355.223680.

Wallace S. Lages and Doug A. Bowman. Walking with adaptive augmented reality workspaces: Design and usage patterns. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 356–366, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. DOI: 10.1145/3301275.3302278. URL

https://doi.org/10.1145/3301275.3302278.

Ricardo Langner, John Brosz, Raimund Dachselt, and Sheelagh Carpendale. Physicsbox: Playful educational tabletop games. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 273–274, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450303996. DOI: 10.1145/1936652.1936712. URL https://doi.org/10.1145/1936652.1936712.

Talia Lavie and Noam Tractinsky. Assessing dimensions of perceived visual aesthetics of web sites. *International Journal of Human-Computer Studies*, 60(3):269 – 298, 2004. ISSN 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2003.09.002. URL http://www.sciencedirect.com/science/article/pii/S1071581903001642.

DoYoung Lee, Jiwan Kim, and Ian Oakley. Fingertext: Exploring and optimizing performance for wearable, mobile and one-handed typing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450380966. DOI: 10.1145/3411764.3445106. URL https://doi.org/10.1145/3411764.3445106.

Johnny C. Lee, Scott E. Hudson, Jay W. Summet, and Paul H. Dietz. Moveable interactive projected displays using projector based tracking. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, page 63–72, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595932712. DOI: 10.1145/1095034.1095045. URL https://doi.org/10.1145/1095034.1095045.

Joon Hyub Lee, Donghyeok Ma, Haena Cho, and Seok-Hyung Bae. 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*, CHI EA '21, New York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450380959. DOI: 10.1145/3411763.3451786. URL https://doi.org/10.1145/3411763.3451786.

Yong Yi Lee, Jong Hun Lee, Bilal Ahmed, Moon Gu Son, and Kwan H. Lee. A new projection-based exhibition system for a museum. *J. Comput. Cult. Herit.*, 12(2), may 2019. ISSN 1556-4673. DOI: 10.1145/3275522. URL https://doi.org/10.1145/3275522.

Sylvain Lefebvre, Samuel Hornus, and Fabrice Neyret. Texture Sprites: Texture Elements Splatted on Surfaces. In *Symposium on Interactive 3D Graphics and Games*, Washington, United States, 2005. ACM SIGGRAPH, ACM Press. URL https://hal.inria.fr/inria-00510158.

Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2): 126–160, June 1994. ISSN 1073-0516. DOI: 10.1145/180171.180173. URL

https://doi.org/10.1145/180171.180173.

W. Lidwell, K. Holden, and J. Butler. *Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design*. Rockport Publishers, 2010. ISBN 9781610580656. URL https://books.google.fr/books?id=l0QPECGQySYC.

Robert W. Lindeman, John L. Sibert, and James K. Hahn. Towards usable vr: An empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 64–71, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 0201485591. DOI: 10.1145/302979.302995. URL https://doi.org/10.1145/302979.302995.

David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 147–160, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368162. DOI: 10.1145/3332165.3347945. URL https://doi.org/10.1145/3332165.3347945.

Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. Design and evaluation of interactive small multiples data visualisation in immersive spaces. pages 588–597, 03 2020a. DOI: 10.1109/VR46266.2020.00081.

Jie Liu, Tim Dwyer, Guido Tack, Samuel Gratzl, and Kim Marriott. Supporting the problem-solving loop: Designing highly interactive optimisation systems, 2020b. URL https://arxiv.org/abs/2009.03163.

Feiyu Lu and Yan Xu. Exploring spatial ui transition mechanisms with head-worn augmented reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. DOI: 10.1145/3491102.3517723. URL https://doi.org/10.1145/3491102.3517723.

Qiuyu Lu, Danqing Shi, Yingqing Xu, and Haipeng Mi. Metalife: Interactive installation based on liquid metal deformable interfaces. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–4, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368193. DOI: 10.1145/3334480.3383134. URL https://doi.org/10.1145/3334480.3383134.

Paul Lubos, Gerd Bruder, Oscar Ariza, and Frank Steinicke. Touching the sphere: Leveraging joint-centered kinespheres for spatial user interaction. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, SUI '16, page 13–22, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340687. DOI: 10.1145/2983310.2985753. URL https://doi.org/10.1145/2983310.2985753.

Weizhou Luo, Anke Lehmann, Hjalmar Widengren, and Raimund

Dachselt. 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*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. DOI: 10.1145/3491102.3501946. URL https://doi.org/10.1145/3491102.3501946.

Ran Ran Man, Dong Sheng Zhou, and Qiang Zhang. A survey of collision detection. In *Mechanical, Electronic and Engineering Technologies (ICMEET 2014)*, volume 538 of *Applied Mechanics and Materials*, pages 360–363. Trans Tech Publications Ltd, 6 2014. DOI: 10.4028/www.scientific.net/AMM.538.360.

Harry M. Markowitz and Alan S. Manne. On the solution of discrete programming problems. *Econometrica*, 25(1):84–110, 1957. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1907744.

R. Marler and Jasbir Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26: 369–395, 04 2004. DOI: 10.1007/s00158-003-0368-6.

Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loic Barthe. Constrained palette-space exploration. *ACM Trans. Graph.*, 36(4), jul 2017a. ISSN 0730-0301. DOI: 10.1145/3072959.3073650. URL https://doi.org/10.1145/3072959.3073650.

Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loic Barthe. Constrained palette-space exploration. *ACM Trans. Graph.*, 36(4), July 2017b. ISSN 0730-0301. DOI: 10.1145/3072959.3073650. URL https://doi.org/10.1145/3072959.3073650.

Jan Meskens, Jo Vermeulen, Kris Luyten, and Karin Coninx. Gummy for multi-platform user interface designs: Shape me, multiply me, fix me, use me. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, page 233–240, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581415. DOI: 10.1145/1385569.1385607. URL https://doi.org/10.1145/1385569.1385607.

Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351, 01 1994. DOI: 10.1117/12.197321.

Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface 2010*, GI '10, page 261–268, CAN, 2010. Canadian Information Processing Society. ISBN 9781568817125.

Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, m. c. schraefel, and Jacob O. Wobbrock. Reducing

legacy bias in gesture elicitation studies. *Interactions*, 21(3):40–45, may 2014. ISSN 1072-5520. DOI: 10.1145/2591689. URL https://doi.org/10.1145/2591689.

M. Müller. Hierarchical position based dynamics. In *VRIPHYS*, 2008.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109 – 118, 2007. ISSN 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2007.01.005. URL http://www.sciencedirect.com/science/article/pii/S1047320307000065.

Miguel A. Nacenta, Satoshi Sakurai, Tokuo Yamaguchi, Yohei Miki, Yuichi Itoh, Yoshifumi Kitamura, Sriram Subramanian, and Carl Gutwin. E-conic: A perspective-aware interface for multi-display environments. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, page 279–288, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936790. DOI: 10.1145/1294211.1294260. URL https://doi.org/10.1145/1294211.1294260.

Ali Neshati, Bradley Rey, Ahmed Shariff Mohommed Faleel, Sandra Bardot, Celine Latulipe, and Pourang Irani. Bezelglide: Interacting with graphs on smartwatches with minimal screen occlusion. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. DOI: 10.1145/3411764.3445201. URL https://doi.org/10.1145/3411764.3445201.

David Chek Ling Ngo, Lian Seng Teo, and John G. Byrne. Modelling interface aesthetics. *Inf. Sci.*, 152(1):25–46, June 2003. ISSN 0020-0255. DOI: 10.1016/S0020-0255(02)00404-8. URL https://doi.org/10.1016/S0020-0255(02)00404-8.

Aziz Niyazov, Nicolas Mellado, Loic Barthe, and Marcos Serrano. Dynamic decals: Pervasive freeform interfaces using constrained deformable graphical elements. *Proc. ACM Hum.-Comput. Interact.*, 5(ISS), nov 2021. DOI: 10.1145/3488538. URL https://doi.org/10.1145/3488538.

Aziz Niyazov, Barrett Ens, Kadek Ananta Satriadi, Nicolas Mellado, Loic Barthe, Tim Dwyer, and Marcos Serrano. User-driven constraints for layout optimisation in augmented reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. DOI: 10.1145/3544548.3580873. URL https://doi.org/10.1145/3544548.3580873.

Donald A. Norman and Stephen W. Draper. User centered system design: New perspectives on human-computer interaction. 1988.

Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings*

of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, page 1233–1244, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450333627. DOI: 10.1145/2858036.2858250. URL https://doi.org/10.1145/2858036.2858250.

Stephen Oney, Brad Myers, and Joel Brandt. ConstraintJS. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, page 229, New York, New York, USA, 2012. ACM Press. ISBN 9781450315807. DOI: 10.1145/2380116.2380146. URL http://dl.acm.org/citation.cfm?doid=2380116.2380146.

L. Ortega and F. Feito. Collision detection using polar diagrams. *Computers & Graphics*, 29(5):726 – 737, 2005. ISSN 0097-8493. DOI: https://doi.org/10.1016/j.cag.2005.08.026. URL http://www.sciencedirect.com/science/article/pii/S0097849305001354.

Brice Parilusyan, Marc Teyssier, Valentin Martinez-Missir, Clément Duhart, and Marcos Serrano. Sensurfaces: A novel approach for embedded touch sensing on everyday surfaces. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(2), jul 2022. DOI: 10.1145/3534616. URL https://doi.org/10.1145/3534616.

J. S. Park, C. Park, and D. Manocha. Efficient probabilistic collision detection for non-convex shapes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1944–1951, 2017.

A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.

Gary Perelman, Emmanuel Dubois, Alice Probst, and Marcos Serrano. 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), nov 2022. DOI: 10.1145/3567738. URL https://doi.org/10.1145/3567738.

Claudio S. Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, page 315–331, Berlin, Heidelberg, 2001. Springer-Verlag. ISBN 3540426140.

Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 955–960, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319522. DOI: 10.1145/2468356.2468527. URL https://doi.org/10.1145/2468356.2468527.

Juri Platonov, Hauke Heibel, Peter Meier, and Bert Grollmann. A mobile markerless ar system for maintenance and repair. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, page 105–108, USA, 2006. IEEE Computer Society. ISBN

1424406501. DOI: 10.1109/ISMAR.2006.297800. URL https://doi.org/10.1109/ISMAR.2006.297800.

G. Michael Poor, Samuel D. Jaffee, Laura Marie Leventhal, Jordan Ringenberg, Dale S. Klopfer, Guy Zimmerman, and Brandi A. Klein. Applying the norman 1986 user-centered model to post-wimp uis: Theoretical predictions and empirical outcomes. *ACM Trans. Comput.-Hum. Interact.*, 23(5), oct 2016. ISSN 1073-0516. DOI: 10.1145/2983531. URL https://doi.org/10.1145/2983531.

Vahid Pooryousef, Maxime Cordeil, Lonni Besançon, Christophe Hurter, Tim Dwyer, and Richard Bassed. Working with forensic practitioners to understand the opportunities and challenges for mixed-reality digital autopsy. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. DOI: 10.1145/3544548.3580768. URL https://doi.org/10.1145/3544548.3580768.

Ivan Poupyrev, Henry Newton-Dunn, and Olivier Bau. D20: Interaction with multifaceted display devices. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, page 1241–1246, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932984. DOI: 10.1145/1125451.1125683. URL https://doi.org/10.1145/1125451.1125683.

Arnaud Prouzeau, Yuchen Wang, Barrett Ens, Wesley Willett, and Tim Dwyer. Corsican twin: Authoring in situ augmented reality visualisations in virtual reality. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375351. DOI: 10.1145/3399715.3399743. URL https://doi.org/10.1145/3399715.3399743.

Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. Scalar: Authoring semantically adaptive augmented reality experiences in virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. DOI: 10.1145/3491102.3517665. URL https://doi.org/10.1145/3491102.3517665.

Roman Rädle, Hans-Christian Jetter, Mario Schreiner, Zhihao Lu, Harald Reiterer, and Yvonne Rogers. 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*, CHI '15, page 3913–3922, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. DOI: 10.1145/2702123.2702287. URL https://doi.org/10.1145/2702123.2702287.

Raf Ramakers, Johannes Schöning, and Kris Luyten. Paddle: Highly deformable mobile devices with physical controls. In *Proceedings of*

the SIGCHI Conference on Human Factors in Computing Systems, CHI '14, page 2569–2578, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324731. DOI: 10.1145/2556288.2557340. URL https://doi.org/10.1145/2556288.2557340.

Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 179–188, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 0897919998. DOI: 10.1145/280814.280861. URL https://doi.org/10.1145/280814.280861.

Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs. Multi-projector displays using camera-based registration. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, VIS '99, page 161–168, Washington, DC, USA, 1999. IEEE Computer Society Press. ISBN 078035897X.

A. Ricci. A Constructive Geometry for Computer Graphics. *computer journal*, 16(2):157–160, May 1973.

Andrew D Rice and Jonathan W Lartigue. Touch-level model (tlm) evolving klm-goms for touchscreen and mobile devices. In *Proceedings of the 2014 ACM Southeast regional conference*, pages 1–6, 2014.

Jan Riemann, Martin Schmitz, Alexander Hendrich, and Max Mühlhäuser. Flowput: Environment-aware interactivity for tangible 3d objects. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), March 2018. DOI: 10.1145/3191763. URL https://doi.org/10.1145/3191763.

Quentin Roy, Futian Zhang, and Daniel Vogel. Automation accuracy is good, but high controllability may be better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–8, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. DOI: 10.1145/3290605.3300750. URL https://doi.org/10.1145/3290605.3300750.

M-A Sabin. The use of potential surfaces for numerical geometry. In *Tech. Report VTO/MS/153, British Aerospace Corp., Weybridge, U.K.*, 1968.

Houssem Saidi, Emmanuel Dubois, and Marcos Serrano. 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*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. DOI: 10.1145/3411764.3445255. URL https://doi.org/10.1145/3411764.3445255.

Houssem Saidi, Laetitia Carreteros, Stephanie Rey, Laurent Truscello, and Youssef Miloudi. Bl.mixedr: Augmenting traditional maintenance procedures to better exploit the capabilities of head-worn ar. In *Pro-*

ceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA '22, page 176–184, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450396318. DOI: 10.1145/3529190.3529210. URL https://doi.org/10.1145/3529190.3529210.

Kadek Ananta Satriadi, Barrett Ens, Maxime Cordeil, Tobias Czauderna, and Bernhard Jenny. Maps around me: 3d multiview layouts in immersive spaces. *Proc. ACM Hum.-Comput. Interact.*, 4(ISS), nov 2020. DOI: 10.1145/3427329. URL https://doi.org/10.1145/3427329.

Ryan Schmidt, Cindy Grimm, and Brian Wyvill. Interactive decal compositing with discrete exponential maps. *ACM Trans. Graph.*, 25(3): 605–613, July 2006. ISSN 0730-0301. DOI: 10.1145/1141911.1141930. URL https://doi.org/10.1145/1141911.1141930.

Greg W. Scragg. User friendly interfaces for data base systems. In *Proceedings of the ACM '82 Conference*, ACM '82, page 158, New York, NY, USA, 1982. Association for Computing Machinery. ISBN 0897910850. DOI: 10.1145/800174.809787. URL https://doi.org/10.1145/800174.809787.

Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 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*, MobileHCI '15, page 161–171, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336529. DOI: 10.1145/2785830.2785838. URL https://doi.org/10.1145/2785830.2785838.

Marcos Serrano, Anne Roudaut, and Pourang Irani. Investigating text legibility on non-rectangular displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 498–508, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. DOI: 10.1145/2858036.2858057. URL http://doi.acm.org/10.1145/2858036.2858057.

Marcos Serrano, Anne Roudaut, and Pourang Irani. Visual composition of graphical elements on non-rectangular displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4405–4416, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. DOI: 10.1145/3025453.3025677. URL http://doi.acm.org/10.1145/3025453.3025677.

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

Florine Simon, Anne Roudaut, Pourang Irani, and Marcos Serrano. Finding information on non-rectangular interfaces. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages

102:1–102:8, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2. DOI: 10.1145/3290605.3300332. URL http://doi.acm.org/10.1145/3290605.3300332.

S. Sistare. Graphical interaction techniques in constraint-based geometric modeling. In *Proceedings of Graphics Interface '91*, GI '91, pages 85–92, Toronto, Ontario, Canada, 1991. Canadian Man-Computer Communications Society. ISBN 0-9695338-0-2. URL http://graphicsinterface.org/wp-content/uploads/gi1991-12.pdf.

Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. User interface façades: Towards fully adaptable user interfaces. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, page 309–318, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933131. DOI: 10.1145/1166253.1166301. URL https://doi.org/10.1145/1166253.1166301.

Ivan E. Sutherland. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, DAC '64, page 6.329–6.346, New York, NY, USA, 1964. Association for Computing Machinery. ISBN 9781450379328. DOI: 10.1145/800265.810742. URL https://doi.org/10.1145/800265.810742.

Ivan E Sutherland et al. The ultimate display. In *Proceedings of the IFIP Congress*, volume 2, pages 506–508. New York, 1965.

Amanda Swearngin, Chenglong Wang, Alannah Oleson, James Fogarty, and Amy J. Ko. Scout: Rapid exploration of interface layout alternatives through high-level design constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. DOI: 10.1145/3313831.3376593. URL https://doi.org/10.1145/3313831.3376593.

Aurelien Tabard, Simon Gurn, Andreas Butz, and Jakob Bardram. A case study of object and occlusion management on the elabbench, a mixed physical/digital tabletop. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, page 251–254, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322713. DOI: 10.1145/2512349.2512794. URL https://doi.org/10.1145/2512349.2512794.

Adam Tapal, Ela Oren, Reuven Dar, and Baruch Eitam. 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. ISSN 1664-1078. DOI: 10.3389/fpsyg.2017.01552. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2017.01552.

John Tinnell. From wimp to atlas: Rhetorical figures of ubiquitous computing. *Computational Culture 5 (15th Jan-*

uary 2016), 2016. URL http://computationalculture.net/from-wimp-to-atlas-rhetorical-figures-of-ubiquitous-computing/.

Kashyap Todi, Daryl Weir, and Antti Oulasvirta. Sketchplore: Sketch and explore with a layout optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, page 543–555, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340311. DOI: 10.1145/2901790.2901817. URL https://doi.org/10.1145/2901790.2901817.

Theophanis Tsandilas. Fallacies of agreement: A critical review of consensus assessment methods for gesture elicitation. *ACM Trans. Comput.-Hum. Interact.*, 25(3), jun 2018. ISSN 1073-0516. DOI: 10.1145/3182168. URL https://doi.org/10.1145/3182168.

Andries van Dam. Post-wimp user interfaces. *Commun. ACM*, 40(2): 63–67, feb 1997. ISSN 0001-0782. DOI: 10.1145/253671.253708. URL https://doi.org/10.1145/253671.253708.

Radu-Daniel Vatavu and Jacob O. Wobbrock. 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*, CHI '15, page 1325–1334, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. DOI: 10.1145/2702123.2702223. URL https://doi.org/10.1145/2702123.2702223.

Roel Vertegaal and Ivan Poupyrev. Introduction. *Commun. ACM*, 51(6): 26–30, jun 2008. ISSN 0001-0782. DOI: 10.1145/1349026.1349033. URL https://doi.org/10.1145/1349026.1349033.

Daniel Vogel and Ravin Balakrishnan. Occlusion-aware interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 263–272, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. DOI: 10.1145/1753326.1753365. URL https://doi.org/10.1145/1753326.1753365.

Manuela Waldner, Raphael Grasset, Markus Steinberger, and Dieter Schmalstieg. Display-adaptive window management for irregular surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, page 222–231, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308717. DOI: 10.1145/2076354.2076394. URL https://doi.org/10.1145/2076354.2076394.

Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 552–567, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386357. DOI: 10.1145/3472749.3474769. URL https://doi.org/

10.1145/3472749.3474769.

Colin Ware. *Information Visualization: Perception for Design: Second Edition*. 04 2004.

Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, jul 1999. ISSN 1559-1662. DOI: 10.1145/329124.329126. URL https://doi.org/10.1145/329124.329126.

Michael Williams, Hassen Gharbi, Aybike Ulusan, Ozlem Ergun, Zhu Xiaofeng, Shiyu Zhang, and Casper Harteveld. Toward human in the loop optimization through game-based experiments. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY Companion '16, page 351–358, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344586. DOI: 10.1145/2968120.2987733. URL https://doi.org/10.1145/2968120.2987733.

Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, page 67–76, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939753. DOI: 10.1145/1449715.1449728. URL https://doi.org/10.1145/1449715.1449728.

Raphael Wimmer, Fabian Hennecke, Florian Schulz, Sebastian Boring, Andreas Butz, and Heinrich Hußmann. Curve: Revisiting the digital desk. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, page 561–570, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589343. DOI: 10.1145/1868914.1868977. URL https://doi.org/10.1145/1868914.1868977.

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

R. S. Woodworth. Sensation and perception: Sensation and perception in the history of experimental psychology. by edwin g. boring. xv + 644 pp. new york: D. appleton-century company, 1942. *Science*, 96(2481):64–65, 1942. DOI: 10.1126/science.96.2481.64. URL https://www.science.org/doi/abs/10.1126/science.96.2481.64.

Brian Wyvill, Andrew Guy, and Eric Galin. Extending the csg tree - warping, blending and boolean operations in an implicit surface modeling system. *Comput. Graph. Forum*, 18(2):149–158, 1999.

Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data Structure for Soft Objects. *The Visual Computer*, 2(4):227–234, February 1986.

Xuhai Xu, Alexandru Dancu, Pattie Maes, and Suranga Nanayakkara. 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*, MobileHCI '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450358989. DOI: 10.1145/3229434.3229449. URL https://doi.org/10.1145/3229434.3229449.

Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. Steering through sequential linear path segments. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 232–243, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. DOI: 10.1145/3025453.3025836. URL https://doi.org/10.1145/3025453.3025836.

Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. 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*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. DOI: 10.1145/3173574.3173652. URL https://doi.org/10.1145/3173574.3173652.

Naoki Yanagihara, Buntarou Shizuki, and Shin Takahashi. 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*, VRST '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370011. DOI: 10.1145/3359996.3364814. URL https://doi.org/10.1145/3359996.3364814.

Yeonsoo Yang and Scott R. Klemmer. Aesthetics matter: Leveraging design heuristics to synthesize visually satisfying handheld interfaces. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, page 4183–4188, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605582474. DOI: 10.1145/1520340.1520637. URL https://doi.org/10.1145/1520340.1520637.

Lei Zhang, Ashutosh Agrawal, Steve Oney, and Anhong Guo. Vrgit: A version control system for collaborative content creation in virtual reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. DOI: 10.1145/3544548.3581136. URL https://doi.org/10.1145/3544548.3581136.