THESIS PRESENTED

TO AWARD THE DEGREE OF

DOCTOR OF

THE UNIVERSITÉ OF BORDEAUX

DOCTORAL SCHOOL OF MATHEMATICS AND COMPUTER SCIENCE

SPECIALTY: COMPUTER SCIENCE

By Rajkumar DARBAR

Extending Interaction Space in Augmented Reality: Contributions in Optical-See-Through and Projection-Based Augmented Environments

Under the supervision of : Martin Hachet
(co-supervisor : Pascal Guitton)

Thesis defense on : 16 SEPTEMBER 2021

Members of the jury:

M. Marcos SERRANO,	Associate Professor	Université de Toulouse	Rapporteur
M. Gilles BAILLY,	Chargé de Recherche	CNRS	Rapporteur
Mme. Caroline APPERT,	Directrice de Recherche	Université Paris Saclay	Examinateur and Président
Mme. Céline COUTRIX,	Chargée de Recherche	CNRS	Examinateur
M. Martin HACHET,	Directeur de Recherche	Inria	Directeur de Thèse
M. Pascal GUITTON,	Professeur	Université de Bordeaux	Co-Directeur

Acknowledgements

First, I would like to sincerely thank my two advisors, Martin Hachet and Pascal Guitton, for supervising this research and encouraging me throughout this journey.

This thesis would not have been possible without exceptional help from Thibault Lainé, Joan Sol Roo, and Arnaud Prouzeau. Thus I would like to especially thank them for their invaluable technical suggestions during my Ph.D.

I would also like to thank in advance all my Ph.D. committee members for taking the time to read this manuscript and traveling to Bordeaux to participate physically in my defense during this exceptional COVID period.

Thank you to all those great minds that I met at POTIOC and other teams from Inria Bordeaux. Special thanks to Patrick Reuter and Fabien Lotte for discussing endless things, from research careers to life in general.

Finally, I am forever grateful to my parents, brother, friends, and other family members for their wonderful support throughout my life.

Abstract

Unlike desktop computing, Augmented reality (AR) blurs the boundaries between the physical and digital world by superimposing virtual information to the real environment. It is possible to create an immersive AR experience either by wearing a head-mounted display or using a projector. This thesis explores interaction challenges associated with these two types of augmented reality displays.

Head-mounted AR displays (AR-HMDs) are constantly improving in terms of display (i.e., the field of view), tracking, interaction techniques, and portability. Currently available input techniques (such as hand-tacking, head/eye-gaze, and voice) in AR glasses are relatively easy to use for some tasks (e.g., grasping and moving an object). However, they lack precision and are not suitable for prolonged usage. Therefore, tasks that require accuracy become difficult to perform. In this research, we consider one such task — text selection that needs character level precision.

On the other hand, projection-based AR, usually called as spatial augmented reality (SAR), augments physical objects directly with digital content using projectors. In SAR, digital augmentation is generally pre-defined, and the user often acts as a passive viewer. A way to enhance the interactivity in SAR environment is to make graphical widgets (pop-up windows, menus, labels, interactive tools, etc.) more accessible. Unfortunately, embedding those widgets within the SAR scene is challenging.

In this dissertation, we explored new interaction techniques to address those challenges mentioned above. Our two main contributions are —

First, we investigated the use of a smartphone as an interactive controller for selecting text in AR displays. We developed four eyes-free, one-handed text selection techniques for AR-HMDs using a smartphone: continuous touch (where smartphone touchscreen acted as a trackpad pad), discrete touch (where smartphone touchscreen was used to move the cursor character by character, word by word, and line by line), spatial movement (smartphone was used as an air-mouse), and raycasting (smartphone was used as a laser pointer). We also compared them in a user study.

Second, we extended the physical space of SAR by providing 2D graphical widgets in mid-air using projection on a drone-mounted panel. Users were able to control the drone position and interact with the projected information dynamically with a handheld controller. We present three possible ways to embed widgets using a drone in SAR: displaying annotations in mid-air, providing interactive tools, supporting different viewpoints. We also describe the implementation details of our approach.

These explorations aim at extending the interaction space in immersive AR applications.

Résumé étendu en Français

Les environnements informatiques de bureau reposent sur un affichage des données au travers d'un écran monoscopique 2D. L'idée de sortir de ce paradigme d'affichage et d'interaction WIMP standard (fenêtres, icônes, menus et pointeur), et de mélanger les espaces physiques qui nous entourent avec des informations numériques, a vu le jour en 1965 lorsqu'Ivan Sutherland a décrit sa vision de l'affichage ultime : "*L'affichage ultime serait, bien sûr, une pièce dans laquelle l'ordinateur peut contrôler l'existence de la matière. [...] Avec une programmation appropriée, un tel affichage pourrait littéralement être le pays des merveilles dans lequel Alice est entrée*". Trois ans plus tard, il présentait le tout premier casque de réalité augmentée (AR-HMD), qui constituait un premier pas vers cet écran ultime [1]. Depuis le "*Sword of Damocles*" de Sutherland, nous avons beaucoup progressé en termes de techniques d'affichage, de suivi et d'interaction au cours des cinq dernières décennies. Récemment, les casques optiques où l'augmentation se fait par transparence (comme HoloLens 2, Magic Leap 1, Nreal Light) sont sortis des laboratoires de recherche pour devenir des objets grand public.

Dans le passé, les dispositifs de RA étaient surtout utilisés pour effectuer des tâches très spécifiques (qui impliquaient principalement la sélection et la manipulation d'objets 3D, la visualisation de données immersives, etc.) dans des domaines d'application spécifiques tels que la réparation et la maintenance, la formation, les jeux, les opérations d'entrepôt, les soins de santé, etc. Cependant, on peut s'attendre à ce que nous utilisions ce type d'affichage pour l'informatique à usage général à l'avenir. En gardant cela à l'esprit, les chercheurs ont récemment commencé à explorer l'utilisation de casques de RA pour le travail impliquant plusieurs fenêtres de travail. En particulier, pour ce type de travail, les fenêtres virtuelles en 2D des casques contiennent souvent des informations textuelles (par exemple, un document PDF, une feuille Excel ou une navigation sur le Web), et les utilisateurs doivent fréquemment effectuer des opérations de saisie et d'édition de texte [2]. Lorsqu'ils sont à leur bureau, ils peuvent profiter d'un clavier standard et d'une souris/trackpad pour effectuer ces tâches efficacement. Mais lorsqu'ils sont en déplacement, la saisie et l'édition de texte ne sont pas aisées car il n'existe pas de techniques de saisie équivalentes au clavier et à la souris/trackpad. Récemment, la saisie de texte pour les casques de RA a attiré l'attention de la communauté des chercheurs [3–6]. En revanche, les recherches liées à l'édition du texte qu'un utilisateur a déjà tapé restent rares [7]. L'édition de texte est une tâche complexe, et la première étape consiste à sélectionner le texte pour l'éditer. Par conséquent, dans la première partie de cette thèse, nous nous concentrerons sur cette partie de sélection du texte.

Au-delà des casques optiques de RA, il est également possible d'augmenter notre espace physique directement avec du contenu numérique en utilisant des projecteurs.

Cette technique est connue sous le nom de réalité augmentée spatiale (RAS). Dans la RAS, nous pouvons utiliser un seul projecteur statique ou orientable, ou plusieurs projecteurs pour augmenter la surface d'affichage potentielle et améliorer la qualité de l'image. Contrairement aux casques de RA, les affichages au travers de projecteurs sont toujours détachés des utilisateurs et intégrés dans les environnements en RAS. Cela permet à plusieurs utilisateurs de visualiser la même augmentation avec des indices de profondeur naturels (à l'exception de la projection stéréoscopique dépendante de la vue, qui est généralement réglée pour un utilisateur unique). Ils peuvent toucher et interagir directement avec les surfaces physiques, ce qui fournit un retour haptique passif et améliore considérablement leur compréhension des informations présentées. En outre, ils peuvent se déplacer librement autour de la maquette physique et découvrir les augmentations sous différents angles de vue et positions. Les utilisateurs peuvent lire les expressions faciales des autres pendant les tâches collaboratives car leurs visages ne sont plus couverts par des casques. De plus, la collaboration n'est pas limitée par le nombre de casques disponibles et interconnectés, mais plutôt par le nombre de personnes qui peuvent être accueillies dans l'espace disponible. Malheureusement, l'une des principales limites des environnements SAR est que le contenu visuel ne peut être affiché que sur des supports physiques. Par conséquent, l'intégration de widgets d'interface utilisateur tels que les menus et les fenêtres contextuelles dans la RAS devient un défi. Ces widgets doivent être positionnés sur les objets physiques augmentés, ce qui entraîne un encombrement visuel qui affecte l'expérience globale de l'utilisateur. La géométrie et le matériau de la scène physique rendent même parfois impossible l'affichage de widgets lisibles [8]. Par conséquent, dans la deuxième partie de cette thèse, nous examinerons comment intégrer des éléments d'interface utilisateur graphique dans la scène RAS.

1 Contributions de la thèse

Nous avons exploré de nouvelles techniques d'interaction pour relever les défis mentionnés ci-dessus. Globalement, cette recherche comporte deux contributions principales décrites ci-dessous.

Exploration de la sélection de texte à l'aide d'un smartphone dans les casques de RA

En général, la sélection de texte dans les casques RA peut être effectuée à l'aide de diverses modalités d'entrée, notamment le suivi de la main, le regard et la visée de la tête, les commandes vocales [7] et les manettes de contrôle. Cependant, ces techniques ont leurs limites. Par exemple, le suivi de la main ne permet pas d'atteindre la précision d'un caractère [9], manque de retour haptique [10], et provoque la fatigue du bras [11]

pendant une interaction prolongée. Le regard et la visée de la tête souffrent du problème du 'Midas Touch', qui provoque l'activation involontaire de commandes en l'absence d'un mécanisme de sélection approprié [12–15]. De plus, les mouvements fréquents de la tête dans l'interaction tête-regard augmentent le mal des transports [16]. L'interaction vocale peut ne pas être socialement acceptable dans les lieux publics [17], et elle peut perturber le flux de communication lorsque plusieurs utilisateurs collaborent. Dans le cas d'un contrôleur portable dédié, les utilisateurs doivent toujours transporter du matériel spécifique supplémentaire.

Récemment, des chercheurs ont exploré l'utilisation d'un smartphone comme entrée pour les casques de RA en raison de sa disponibilité (il peut même être l'unité de traitement des casques [18]), de sa familiarité, de son acceptabilité sociale et de son aspect tangible [19–21]. Il ne fait aucun doute qu'il existe un énorme potentiel pour la conception de nouvelles applications inter-appareils avec la combinaison d'un écran RA et d'un smartphone. Dans le passé, les smartphones ont été utilisés pour interagir avec différentes applications fonctionnant sur des casques, comme la manipulation d'objets 3D [22], la gestion de fenêtres [23], la sélection de menus graphiques [24] et ainsi de suite. Cependant, nous n'avons pas connaissance d'une recherche ayant étudié la sélection de texte dans un affichage RA à l'aide d'un smartphone disponible dans le commerce. Par conséquent, nous proposons quatre techniques de sélection de texte sans contact visuel pour les casques de RA utilisant un smartphone comme contrôleur d'entrée — le toucher continu, le toucher discret, le mouvement spatial et le rayon virtuel.

Pour sélectionner du texte avec succès en utilisant l'une des techniques proposées, l'utilisateur doit suivre la même séquence d'étapes à chaque fois. Tout d'abord, il déplace le curseur, situé dans la fenêtre de texte d'un écran de RA, jusqu'au début du texte à sélectionner (c'est-à-dire le premier caractère). Ensuite, il effectue un double tapotement sur le téléphone pour confirmer la sélection de ce premier caractère. Il peut voir sur l'écran du casque que le premier caractère a été mis en évidence en jaune. En même temps, elle passe en mode de sélection de texte. Ensuite, il continue à déplacer le curseur jusqu'à la position finale du texte en utilisant l'une des techniques présentées ci-dessous. Pendant que le curseur se déplace, le texte est également mis en évidence simultanément jusqu'à la position actuelle du curseur. Enfin, il termine la sélection du texte par un second double-tap.

Continuous Touch: Cette technique utilise le smartphone comme un trackpad (voir Figure. 3.10(a)). Il s'agit d'une technique de pointage indirect où l'utilisateur déplace son pouce sur l'écran tactile pour changer la position du curseur sur l'affichage RA. Pour la mise en correspondance entre l'affichage et l'écran tactile, nous avons utilisé un mode relatif avec embrayage. Comme l'embrayage peut dégrader les performances, un gain de contrôle de l'affichage (CD) a été appliqué pour le minimiser.

Discrete Touch: Cette technique s'inspire de la sélection de texte avec les raccourcis clavier disponibles dans les systèmes Mac [25] et Windows [26]. OS. Dans ce travail, nous

avons essayé d'émuler quelques raccourcis clavier. Nous avons particulièrement envisagé d'imiter les raccourcis clavier liés à la sélection de texte au niveau des caractères, des mots et des lignes. Par exemple, sous Mac OS, le fait de maintenir la touche ⇧ enfoncée et d'appuyer sur la touche → ou ← étend la sélection de texte d'un caractère vers la droite ou la gauche. Tandis que maintenir la touche ⇧+⌥ enfoncée et appuyer sur → ou ← permet aux utilisateurs de sélectionner le texte d'un mot vers la droite ou la gauche. Pour sélectionner le texte jusqu'au caractère le plus proche situé à la même position horizontale sur la ligne supérieure ou inférieure, l'utilisateur doit maintenir la touche ⇧ enfoncée et appuyer sur la touche ↑ ou ↓ respectivement. Dans le cadre d'une interaction tactile discrète, nous avons reproduit tous ces raccourcis à l'aide de gestes de balayage directionnel (voir Figure. 3.10(b)). Le glissement vers la gauche ou la droite permet de sélectionner le texte aux deux niveaux - mot et caractère. Par défaut, il fonctionne au niveau du mot. Les utilisateurs effectuent un tap long qui agit comme un bouton de basculement pour passer de la sélection au niveau du mot à celle du caractère. D'autre part, un glissement vers le haut ou vers le bas permet de sélectionner le texte à une ligne au-dessus ou au-dessous de la position actuelle. L'utilisateur ne peut sélectionner respectivement qu'un seul caractère/mot/ligne à la fois avec son geste de balayage.

Mouvement spatial: Cette technique émule le smartphone en tant que souris tenu en l'air [27, 28] pour les casques de RA. Pour contrôler la position du curseur sur l'écran du casque, l'utilisateur tient le téléphone devant son torse, place son pouce sur l'écran tactile, puis déplace le téléphone dans l'air avec de petits mouvements de l'avant-bras dans un plan perpendiculaire à la direction du regard (voir Figure. 3.10(c)). Pendant le déplacement du téléphone, les données de position suivies en coordonnées XY sont traduites en mouvement du curseur en coordonnées XY dans une fenêtre 2D. Lorsqu'un utilisateur souhaite arrêter le mouvement du curseur, il lui suffit de lever son pouce de l'écran tactile. Les événements de toucher et relâcher avec le pouce définissent le début et l'arrêt du mouvement du curseur sur l'écran RA. L'utilisateur détermine la vitesse du curseur en déplaçant simplement le téléphone plus rapidement ou plus lentement en conséquence. Nous avons appliqué un gain de CD entre le mouvement du téléphone et le déplacement du curseur sur la fenêtre de texte.

Raycasting: Le raycasting est une technique d'interaction populaire dans les environnements RA/RV pour sélectionner des objets virtuels 3D [29, 30]. Dans ce travail, nous avons développé une technique de raycasting basée sur un smartphone pour sélectionner du texte affiché sur une fenêtre 2D dans le casque (voir Figure. 3.10(d)). Un smartphone suivi 6 DoF a été utilisé pour définir l'origine et l'orientation du rayon. Dans l'affichage du casque, l'utilisateur peut voir le rayon en ligne droite qui apparaît depuis le haut du téléphone. Par défaut, le rayon est toujours visible pour les utilisateurs du caque de RA tant que le téléphone est correctement suivi. Pour orienter le rayon, l'utilisateur doit effectuer de petits mouvements angulaires du poignet pour pointer le contenu textuel. Lorsque le rayon touche la fenêtre de texte, l'utilisateur y voit le curseur. Par rapport aux autres méthodes proposées, le raycasting ne nécessite pas d'embrayage car il permet de

pointer directement la cible. L'utilisateur confirme la sélection de la cible sur l'affichage RA en fournissant une entrée tactile (c'est-à-dire un double-tap) à partir du téléphone.

Nous avons évalué ces quatre techniques dans le cadre d'une étude auprès de 20 participants où les utilisateurs devaient sélectionner du texte à différents niveaux de granularité. Nos résultats suggèrent que le toucher continu, dans lequel nous avons utilisé le smartphone comme un trackpad, a surpassé les trois autres techniques en termes de temps d'exécution de la tâche, de précision et de préférence des utilisateurs.

Extension des espaces physiques en Réalité Augmentée Spatiale à l'aide d'un drone

Nous avons proposé *DroneSAR* pour fournir des widgets graphiques interactifs en RAS, comme un menu flottant dans l'air, en utilisant la projection sur un panneau monté sur un drone, tandis que les utilisateurs émettent des entrées via un contrôleur manuel pour interagir avec la scène et positionner le drone de manière dynamique. Cette approche présente plusieurs avantages. Premièrement, par rapport à la projection directe sur un objet, la qualité de la projection ne dépend pas de la géométrie et du matériau de la scène physique, ce qui garantit une bonne visualisation des widgets. Deuxièmement, les utilisateurs peuvent se concentrer sur la zone d'intérêt sans diviser leur attention avec une deuxième zone d'interaction (c'est-à-dire un panneau tenu dans la main, une tablette, etc.) et ils peuvent se déplacer librement dans l'espace. Troisièmement, ils peuvent positionner les widgets à des emplacements 3D spécifiques, qui peuvent être éloignés. La technique proposée leur permet de voir les widgets dans leur contexte spatial 3D. Les utilisateurs auront l'impression que le contenu projeté sur le drone est toujours sémantiquement lié aux surfaces physiques augmentées. Enfin, plusieurs utilisateurs sont capables de percevoir la même information en même temps, ce qui favorise le travail collaboratif. Nous avons présenté trois façons possibles d'intégrer des widgets à l'aide d'un drone — afficher des annotations dans l'air, fournir des outils interactifs, soutenir différents points de vue. De nombreuses autres fonctionnalités pourraient être imaginées, où *DroneSAR* permet de faire évoluer des applications de bureau standards vers le domaine des environnements en réalité augmentée spatiale.

Affichage des annotations en l'air: L'ajout d'annotations dans la RAS enrichira l'expérience de l'utilisateur, mais le placement des étiquettes associées au monde physique augmenté n'est pas trivial en raison de sa surface de projection non plane et texturée. Pour résoudre ce problème, *DroneSAR* permet de projeter des annotations virtuelles sur le drone, indépendamment de la surface de projection. Tout en affichant l'étiquette en l'air, les utilisateurs peuvent positionner le drone à côté de l'objet physique à l'aide d'un contrôleur portable pour créer un lien entre l'annotation et la région d'intérêt (ROI) dans l'espace physique. Ils ont également la possibilité de positionner le drone de manière automatique

définie par l'application. En outre, notre système permet aux utilisateurs d'interagir avec ces étiquettes projetées à l'aide des boutons de saisie du contrôleur. S'il s'agit d'un texte ou d'une image, ils peuvent utiliser le trackpad du contrôleur pour modifier son orientation. Dans le cas d'une vidéo, ils peuvent la lire ou la mettre en pause avec le bouton de déclenchement. Pour afficher les étiquettes, nous avons implémenté un widget dédié. Comme décrit dans la Figure 4.13(A), lorsque l'étiquette '*cheminée*' doit être affichée, le drone s'approche automatiquement (c'est-à-dire d'une manière définie par le système) de la cheminée de la maison et y fait du surplace. De même, pour pointer un endroit spécifique dans les airs, nous projetons l'image d'un curseur sur le panneau du drone, et à l'aide du trackpad, les utilisateurs modifient son orientation (voir la Figure 4.13(B)). Enfin, *DroneSAR* peut également afficher une vidéo 2D dans la scène, comme le montre la Figure 4.13(C).

Fournir des outils interactifs: Dans le domaine de la réalité augmentée spatiale, les utilisateurs agissent souvent comme des spectateurs passifs. Il serait intéressant de leur fournir des outils interactifs pour jouer avec l'augmentation virtuelle sur les objets physiques de façon dynamique. Inspirés par les "dynamic shader lamps" [31], nous avons augmenté le panneau du drone avec plusieurs outils virtuels. Les utilisateurs peuvent sélectionner un outil en le pointant à l'aide d'un contrôleur. Une fois sélectionné, le contrôleur devient le mandataire de cet outil et lui permet d'effectuer une opération spécifique sur le contenu augmenté. Par exemple, un utilisateur peut sélectionner un outil de mesure dans le menu principal du panneau du drone, illustré sur la Figure 4.2(A). Comme l'illustre la Figure 4.2(B), les participants tracent une ligne sur la maison augmentée à l'aide du bouton de déclenchement du contrôleur, et la longueur mesurée s'affiche sur le panneau du drone. On peut facilement l'étendre à une application de peinture où le panneau du drone sera enrichi de différents outils (palette de couleurs, coup de pinceau, etc.).

En outre, au lieu de fournir une palette d'outils virtuels, le drone lui-même peut agir comme un proxy pour un outil particulier. En déplaçant le drone à l'aide d'un contrôleur, les utilisateurs accomplissent la fonction de cet outil. Pour illustrer cela, nous fournissons un outil lié à la source lumineuse de la scène. Dans ce cas, le drone agit comme un proxy de la source de lumière virtuelle. Les utilisateurs peuvent modifier de manière interactive la position de la lumière, ce qui serait difficile à réaliser sans le retour d'information sur la position en l'air que fournit le drone. L'apparence de la maison est modifiée en conséquence lorsqu'ils déplacent la lumière de droite à gauche (voir la Figure 4.14(A & B)). On obtient ainsi une visualisation tangible d'un objet non physique qui s'inspire du projet *Urp*[32].

Support de différents points de vue: Une autre caractéristique intéressante de *DroneSAR* est d'afficher une vue 3D interactive de l'objet augmenté observé à proximité de la zone d'intérêt. En effet, même si les environnements RAS présentent divers avantages intéressants, leur caractère physique implique également de fortes limitations par rapport

aux environnements purement virtuels. Il n'est pas possible de voir les objets physiques augmentés depuis une vue de dessus ou de derrière, et l'échelle des objets reste toujours fixe. Inspirés par le concept de *One Reality* [33] qui combine le RAS et la RV pour ajouter de la flexibilité aux mondes physiques, nous proposons une approche où le *DroneSAR* est utilisé comme une visionneuse interactive 3D contextuelle. Les participants peuvent voir l'objet physique augmenté sous différents angles et à différentes échelles en utilisant le trackpad et le bouton de déclenchement du contrôleur tout en restant ancrés dans l'environnement physique. Ainsi, ils peuvent facilement faire le lien entre l'objet réel-augmenté et son homologue virtuel (voir la Figure 4.14(C)).

En résumé, certaines tâches de base (telles que la saisie de texte, l'édition de texte, la sélection de menus, l'interaction avec des widgets graphiques, etc.) sont simples à réaliser sur notre environnement informatique de bureau mais difficiles à accomplir dans un environnement immersif de RA. Pour surmonter ces difficultés, nous avons contribué à améliorer l'espace d'interaction des deux formes les plus courantes de réalité augmentée immersive. Certes, les techniques d'interaction proposées ne sont pas des solutions génériques. Elles dépendent du contexte de l'application. Néanmoins, nous n'avons fait qu'effleurer la surface de ce qui est possible de faire avec un smartphone et un drone dans le contexte de la réalité augmentée immersive.

Contents

Acknowledgements ii

Abstract iii

Résumé étendu en Français iv
1 Contributions de la thèse . . . v
Exploration de la sélection de texte à l'aide d'un smartphone dans les casques de RA . . . v
Extension des espaces physiques en Réalité Augmentée Spatiale à l'aide d'un drone . . . viii

Contents xi

1 Introduction 1
1.1 Research Questions . . . 6
1.2 Research Contributions . . . 7
1.3 Organization of the Thesis . . . 8
1.4 Publications . . . 8

2 Related Work 9
2.1 Interaction Space in AR-HMDs . . . 9
Hand Tracking Based Interaction . . . 9
Head-Gaze Based Interaction . . . 12
Eye-Gaze Based Interaction . . . 13
Body Based Interaction . . . 15
Pen Based Interaction . . . 17
Mobile Devices Based Interaction . . . 20
Multimodal Interaction . . . 23
Extending Display Space . . . 25
2.2 Interaction Space in SAR . . . 27

3 Exploring Smartphone-enabled Text Selection in AR-HMDs 33
3.1 Introduction . . . 33
3.2 Specific Related Work . . . 35
Text Selection and Editing in AR glasses . . . 35
Combining Handheld Devices and Large Wall Displays . . . 36
Text Selection on Handheld Devices . . . 37
3.3 Designing Smartphone-Based Text Selection in AR-HMDs . . . 40
Design Criteria . . . 41
Proposed Techniques . . . 42

Implementation . . . 45
3.4 Experiment . . . 47
Participants and Apparatus . . . 47
Task . . . 48
Study Design . . . 49
Procedure . . . 49
Measures . . . 50
Hypotheses . . . 51
3.5 Result . . . 51
Task Completion Time . . . 52
Error Rate . . . 52
Questionnaires . . . 52
3.6 Discussion & Design Implications . . . 54
3.7 Limitations . . . 57
3.8 Conclusion . . . 58

4 Extending Physical Spaces in SAR using Projection on a Drone 59
4.1 Introduction . . . 59
4.2 Specific Related Work . . . 61
Mid-air Imaging . . . 61
Drone as a Mid-air Display . . . 62
4.3 DroneSAR . . . 66
Displaying Annotations in Mid-air . . . 67
Providing Interactive Tools . . . 68
Supporting Different Viewpoints . . . 69
4.4 Implementation . . . 70
DroneSAR System . . . 70
Tracking System . . . 71
Projector Calibration . . . 71
Drone Hardware . . . 72
Drone Navigation . . . 72
User Interaction . . . 73
4.5 Drone Positioning Evaluation . . . 75
4.6 Limitations . . . 76
4.7 Conclusion . . . 77

5 Conclusion and Future Work 79
5.1 Revisiting Thesis Contributions . . . 79
5.2 Future Work . . . 80
5.3 Concluding Remarks . . . 81

Bibliography 83

List of Figures

1.1 Comparing physical monitors, virtual AR monitors, and a hybrid combination of both for productivity tasks [36]: (a) Physical condition had three monitors side-by-side; (b) Virtual condition had three monitors rendered through HoloLens; (c) Hybrid condition combined a central physical monitor with two peripheral virtual monitors. 2

1.2 (A) activity: doodling, cognitive load: low; (B) activity: brainstorming, cognitive load: medium; (C) activity: reading scientific paper, cognitive load: high. Based on the current context, Mixed Reality interfaces adapt automatically [40]. For example, the interface shows more elements in more detail in the case of a low cognitive load task (A). Whereas increased cognitive load leads to a minimal UI with fewer elements at lower levels of detail (B to C). 3

1.3 (A) In an AR workspace, virtual windows are embedded into the user's nearby surfaces [38]; (B & C) AR interfaces adapt dynamically based on the user's walking and physical environment [41]. 3

1.4 (A) Collaborating through *Augmented Surfaces* [46]; (B) *DigitalDesk* prototype where users selected a printed number on a piece of paper and put that number into the calculator application [47]; (C) *PaperWindows* prototype with three pages [49]. 5

1.5 (A) The physical model of the *Taj Mahal*. (B) This model is illuminated with projectors to simulate different materials [52]. 5

1.6 *IllumiRoom* is a proof-of-concept system that augments the physical environment surrounding a television to enhance interactive experiences [53]. (a) With a 3D scan of the physical environment, we can (b) directly extend the FOV of the game, (c) selectively render scene elements. 6

2.1 A docking task completed with PinNPivot [61]: (a) Source and target. (b) 6DOF manipulation quickly turns the object around. (c) Scaled 3DOF translation accurately places the spout's tip. (d) A pin is created. (e) It is locked. (f) The object is quickly rotated in 3DOF. (g) It is accurately rotated in scaled 2DOF. (h) A second pin is created. (i) It is locked, and a ring appears. (j) The object is rotated in scaled 1DOF. When the target turns yellow, it indicates a good fit. . 10

2.2 Two example gestures for authoring animations [62]: (a-b) a gesture manipulating the direction, spread, and randomness of smoke emission; (c-d) a gesture directly bending an object to describe a follow-through behaviour. 11

2.3 Selected mid-air mode-switching techniques [67]: (A) non dominant fist; (B) non dominant palm; (C) hand in field of view; (D) touch head; (E) dominant fist; (F) dominant palm; (G) point; (H) orientated pinch; (I) middle finger pinch. 11

2.4 MRTouch enables touch interaction in AR-HMDs [72]. (a) When a user approaches a surface, MRTouch detects the surface and (b) presents a virtual indicator. (c) The user touch-drags directly on the surface to (d) create a launcher and start an app. (e) In this app, the user uses touch to precisely rotate a 3D model. 12
2.5 A user is entering text using DepthText in a VR-HMD. The user has entered the word “excellent" by performing short forward movements towards the depth dimension (z-axis) with an acceleration speed larger than threshold a_0 13
2.6 Head gestures set for different commands [77]. The movement of head is indicated by the arrows. “2×" represents the repeating of the action for twice. “1s" is an illustration for a dwell. 13
2.7 Outline Pursuits [85] support selection in occluded 3D scenes. A: The user points at an object of interest, but the selection is ambiguous due to occlusion by other objects. B: Potential targets are outlined, with each outline presenting a moving stimulus that the user can follow with their eye-gaze. C: Matching of the user’s smooth pursuit eye movement completes the selection. Note that outline pursuits can augment manual pointing as shown or support hands-free input using the head or gaze for initial pointing. 15
2.8 Three interaction methods proposed by Liu et al. [86] to manipulate 3D object rotation. 15
2.9 The eyemR-Vis prototype system, showing an AR user (HoloLens2) sharing gaze cues with a VR user (HTC Vive Pro Eye) and vice-versa. 16
2.10 PalmType [92] enables text input for (a) smart glasses using a QWERTY keyboard interface, by using (b) wrist-worn sensors to detect the pointing finger’s position and taps, and (c) displaying a virtual keyboard with highlighted keys via the display of the glasses. 16
2.11 (a) Leveraging the palm as a gesture interface in PalmGesture [93]. (b) Tracking bright regions in the infrared camera image. (c) Drawing an email symbol to check emails on Google Glass. 17
2.12 (A) The menu widget concept on forearm. A user is interacting with the menu using (B) touch (C) drag (D) slide and (E) rotate gestures. 18
2.13 (A) Word-gesture and tap based typing around the thigh [97]. (B) Foot-taps as a direct and indirect input modality for interacting with HMDs [98]. (C) A user needs to go the North-East direction using DMove technique and a selection is made when the user (nearly) completes the action [16]. 18
2.14 Pointing tasks in VR (top) and AR (bottom) using a mouse, a VR controller, and a 3D pen [106]. 19
2.15 A participant is holding the pen with (A) a precision grip and (B) a power grip. 19
2.16 PintAR [108] in use. (A) Designer sketching an interface element on the tablet using a pen. (B) Designer placing sketched element in the environment using an Air tap. (C) AR interface element placed side-by-side with real display. . . 19

2.17 Participants were drawing a stroke around (A) a physical object and (B) a virtual object [110]. 20
2.18 (A) The overall setup of a stylus-based text input system in a CAVE environment [111]. (B) The Tilt-Type interface. (C) The Arc-Type interface. 20
2.19 MultiFi [21] widgets crossing device boundaries based on proxemics dimensions (left), e.g., middle: ring menu on a smartwatch (SW) with head-mounted display (HMD) or right: soft keyboard with full-screen input area on a handheld device and HMD. 21
2.20 The BISHARE [20] design space of joint interactions between a smartphone and augmented reality head-mounted display. Each cell contains a single example of joint interaction, but represents a broader class of interaction techniques that may be possible. 21
2.21 (A) Understanding window management interactions using an AR-HMD + smartphone interface [23]. (B) Enlarging smartphone display with an AR-HMD [112]. 22
2.22 Data visualization using mobile devices and Augmented Reality head-mounted displays [113]: (a) Envisioned usage scenario; (b) 2D scatter-plot extended with superimposed 3D trajectories/paths; (c) 3D wall visualization in AR aligned with the mobile device; (d) Use of AR for seamless display extension around a geographic map; (e) Combining visualizations with an AR view between the devices. 22
2.23 Left: 3D data spaces can be explored by (a) 3D panning and (b) zooming relative to their fixed presentation space. Right: A user wearing a HoloLens explores such a 3D data space with smartphone-based proposed interaction techniques [19]. 23
2.24 Typing on a midair auto-correcting keyboard with word predictions (left) vs. speaking a sentence and then correcting any speech recognition errors (right) [127]. Users correct errors by selecting word alternatives proposed by the speech recognizer or by typing on the virtual keyboard. 24
2.25 Gaze + Pinch interactions unify a user's eye gaze and hand input: look at the target, and manipulate it (a); virtual reality users can utilise free hand direct manipulation (b) to virtual objects at a distance in intuitive and fluid ways (c). 24
2.26 Radi-Eye [130] in a smart home environment for control of appliances. A: The user turns on the lamp via a toggle selection with minimal effort using only gaze (orange) and head (red) movements. B: Selection can be expanded to subsequent head-controlled continuous interaction to adjust the light colour via a slider. C: Gaze-triggered nested levels support a large number of widgets and easy selection of one of the multiple preset lighting modes. The widgets enabled via Radi-Eye allow a high level of hands-free and at-a-distance control of objects from any position. 25

2.27 Extending large interactive display visualizations with Augmented Reality [135]. (A) Two analysts are working on data visualization tasks. (B) Displaying AR Brushing and Linking, Embedded AR Visualizations, and Extended Axis Views. (C) Hinged Visualizations to improve the perception of remote content. (D) Curved AR Screen is providing an overview of the entire display. 26
2.28 ShARe [137] is a modified AR-HMD consisting of a projector and a servo motor attached to its top (a). This allows people in the surrounding to perceive the digital content through projection on a table (b, f) or on a wall (e) and interact via finger-based gestures (c) or marker-based touch (d). 26
2.29 Dynamic shader lamps for applying virtual paint and textures to real objects simply by direct physical manipulation of the object and a “paint brush" stylus. [31]. 27
2.30 Motivating surface interaction examples of the “Build, Map, Play" process proposed by Jones et al. [142]: (A) a virtual miniature golf game, (B) a two-player tank game, (C) a photo viewer. (D) The user selects a menu item with a stylus from a surface adaptive radial menu that adapts to the surface that it is displayed on. 28
2.31 Augmented handheld tool is providing virtual widgets for a paint application [57]. 29
2.32 Examples of ad hoc controls, in which the wooden cube is the handle for slider interaction [144]. (A) Video-editing controls on their own. (B) The same controls supplementing the existing standard controls. 29
2.33 (A) The concept of floor projected UI in a collaborative SAR environment [145]. (B) Experimental setup with the extended floor UI. (C) Participants discussing a virtual scene. (D) The virtual scene from the master’s (the user who controls the perspective) point of view. 30
2.34 MirageTable is a curved projection-based augmented reality system (A), which digitizes any object on the surface (B), presenting correct perspective views accounting for real objects (C) and supporting freehand physics-based interactions (D). 30
2.35 Mobile phone pointing techniques in SAR [147]: (A) viewport, where targets are captured by a camera-like view; (B) raycasting, where targets are pointed at; (C) tangible, where targets are directly contacted. 30
2.36 (A) The user is moving a cursor (represented in blue) to a target (represented in red) on an augmented object using a standard mouse [148]. (B) In DesignAR system, a user is manipulating projected content for interior design using a tablet [58]. 31
2.37 Example scenes to describe One Reality framework [33]: volcano mock-up made out of sand (top), 3D printed Toyota engine (bottom). Each scene can be interacted with different display technologies: spatial augmentation (left), see-through displays (middle), and opaque head mounted displays (right). . . 31

3.1 Force-assisted text selection technique [155]: (a) the evaluation setup; (b-c) a force-assisted button interface on an iPhone 7 to select the textual contents on a distal display. 35
3.2 EYEditor interactions [7]: User sees the text on a smart glass, sentence-by-sentence. In the Re-speaking mode, correction is achieved by re-speaking over the text and a hand-controller is used to navigate between sentences. Users can enter the Select-to-Edit mode to make fine-grained selections on the text and then speak to modify the selected text. 36
3.3 Distant freehand pointing and clicking on a very large, high resolution displays [157]: (A) raycasting (B) relative pointing with clutching (C) hybrid ray-to-relative pointing. 37
3.4 The workflow of using Text Pin to select text [163]. (a) Widget A appears at the point of touch with a magnifying lens displaying above it; (b) Widget A consists of a handle and a circle (the magnifying lens above it disappears once the finger is lifted off the screen); (c) Finger clicks on the circle to fix one end of selection; (d) Widget B appears when finger clicks on the opposite end of selection (note we do not show the magnifying lens above Widget B for a clearer illustration of Widget B); (e) The user clicks on the circle of Widget B and (f) repeats step (a-d) to select the next non-adjacent text. 38
3.5 Some Gedit editing gestures on the smartphone keypad in one-handed use [165]. All gestures start from the right edge: (A) a flick left to select a word; (B) a clockwise ring gesture selects characters to the right of the text cursor; (C) the copy gesture 'C'; (D) the paste gesture 'V'; (E) clockwise and counterclockwise ring gesture for cursor control; (F) the left thumb swipes from the left edge to trigger editing mode, and then editing gestures can be performed by the right thumb; the ring gesture in editing mode performs text selection; the user simply lifts the left thumb to stop editing. 38
3.6 Usage of Press & Tilt technique [166]. 39
3.7 Usage of Press & Slide technique [167]. 39
3.8 Force-sensitive text selection on touch devices [153]. (A) 'mode gauge' — using force for different selection modes (B) Example of text selection using force on a touchscreen. The user performs a long-press that displays the callout magnifier. Keeping the force in the character level, the user adjusts its position by moving her finger. She then presses harder to start the manipulation of the second cursor. If she releases her finger while the force is at the sentence level of the 'mode gauge', she will select the whole paragraph. If she releases her finger while the force is at the character level, she will only select what is between the two cursors. 39
3.9 Illustration of the Gaze'N'Touch concept [170]. (A) Look at the starting character; (B) Touch down; (C) Look at the end character; (D) Release touch. 40
3.10 Illustrations of our proposed interaction techniques: (a) continuous touch; (b) discrete touch; (c) spatial movement; (d) raycasting. 45

3.11 (a) The overall experimental setup consisted of an HoloLens, a smartphone, and an optitrack system. (b) In the HoloLens view, a user sees two text windows. The right one is the 'instruction panel' where the subject sees the text to select. The left is the 'action panel' where the subject performs the actual selection. The cursor is shown inside a green dotted box (for illustration purpose only) on the action panel. For each text selection task, the cursor position always starts from the center of the window. 48
3.12 Text selection tasks used the experiments: (1) word (2) sub-word (3) word to a character (4) four words (5) one sentence (6) paragraph to three sentences (7) one paragraph (8) two paragraphs (9) three paragraphs (10) whole text. 49
3.13 Mean task completion time for our proposed four interaction techniques. Lower scores are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$). 52
3.14 Mean error rate of interaction techniques. Lower scores are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$). 53
3.15 Mean scores for the ranking questionnaire which are in 3 point likert scale. Higher marks are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$). 54
3.16 Mean scores for the NASA-TLX task load questionnaire which are in range of 1 to 10. Lower marks are better, except for performance. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$). 54

4.1 (A) A physical house mock-up. (B) A drone is mounted with two white paper panels. 60
4.2 An example scenario of DroneSAR. (A) The house is augmented using projection, and the main menu is composed of a set of virtual tools projected on the drone panel. (B) A user selected the 'measuring tool' application using a controller. Then, the user positions the drone at the desired location in the 3D space (i.e., on top of the house) and draws a line shown in blue color on the augmented house to measure its width. Finally, the measured length is displayed on the drone panel. 60
4.3 The fog-display system [188]: (A) overall concept and (B) a prototype of it. . . 62
4.4 The water-drop display [191]: (A) overall concept and a prototype setup; (B) the display is showing an image and (C) text. 62
4.5 Pixie Dust [192]: (Top) Hardware setup; (Bottom) Example images created with this setup. 63
4.6 (A) Display drone in an outdoor setting [196]. (B) Drone-projected in-situ navigation instructions [197]. 64

4.7 Different example scenarios of LightAir system [200]: (a) LightAir for human-drone communication (b) DronePiano application (c) 3D point cloud processing for DroneBall. 64
4.8 A working scenario of Drone.io setup [201]. (A) Top view of the projected radial interface as seen by the user. (B) A user is extending his arm in a push gesture to navigate through the menu with the drone flying above. (C) A user is selecting an item in the menu. 64
4.9 (A) iSphere prototype [205]. (B) LightBee [207] telepresence system with two local users viewing the drone-based light field display to communicate with a person in a remote capture room (inset) 65
4.10 ARial Texture tracks the position and orientation of a drone and projects a texture on the drone's propellers accurately [206]. 65
4.11 (A) BitDrones [210] hovering in a tight formation. (B) User inspecting a remote facility with telepresence functionality provided by the drone. (C) User resizing a compound object using a bi-manual pinch gesture by moving drones. 66
4.12 (Left) In SAR, the projection space is limited by the size of the physical object. (Right) DroneSAR extends this projection space (shown in yellow color) with a flying panel that can be positioned in the surround of the physical scene. . . . 66
4.13 (A) The drone is hovering next to the chimney to display its corresponding label. (B) A flying cursor allows participants to point at a specific location in the scene. The dotted circle in the image represents the particular location in mid-air. (C) A video explaining the history is displayed near the medieval house. 68
4.14 (A - B) The light source of our scene is at the drone hovering position. By moving the light source, the user is casting shadows on the scene. (C) An interactive 3D model of the mock-up displayed next to the physical one allows the user to observe the scene from another viewpoint. 70
4.15 Overall architecture of DroneSAR system. 70
4.16 A HTC vive tracker with retro reflective markers. 71
4.17 Drone flight control to reach the goal position. 72
4.18 The referenced scene (A) is decomposed into solid cells (in red) (B), then 'safe' cells (in yellow) (C). Example of way-point cells (in light green) (D). 74
4.19 Positioning the drone at different target locations with respect to each of the four surfaces. Here, **d** represents distance to the target from the surface. 76
4.20 The drone positioning error to the target locations for all four surfaces. Overall error is 6.8 ± 1.1cm. 76

List of Tables

3.1 Logistic function parameter values for continuous touch and spatial movement interaction. The unit of CD_{Max} and CD_{Min} is in mm/mm, whereas λ is in sec/mm and V_{inf} is in mm/sec. 46
3.2 Properties, advantages, and limitations of each input interaction. 57

1 Introduction

In traditional desktop computing, we are limited by a 2D flat window to reach the digital world. The idea of getting out of this standard WIMP (windows, icons, menus, and pointer) interface and augmenting our physical space with digital information has started in 1965, when Ivan Sutherland described his vision of the ultimate display [34] — "*The ultimate display would, of course, be a room within which the computer can control the existence of matter. [...] With appropriate programming, such a display could literally be the Wonderland into which Alice walked*". Three years later, he presented the first-ever head-mounted augmented reality display (AR-HMD) as a very early step towards such an ultimate display [1]. Since Sutherland's Sword of Damocles, we have progressed significantly in terms of display, tracking, and interaction techniques in the last five decades. Recently, optical see-through AR-HMDs (like HoloLens 2, Magic Leap 1, Nreal Light) are emerging from research labs into the mainstream.

In the past, AR devices were mostly used for performing very specific tasks (which mainly involve selecting and manipulating 3D objects, immersive data visualization, etc.) in specific application areas such as repairing and maintenance, training, gaming, warehouse operations, healthcare, etc. However, it is expected that we will use these displays for general-purpose computing in the future [35]. Keeping this in mind, lately, researchers have started exploring AR-HMDs for knowledge work. For example, Pavanatto et al. [36] compared physical monitors, virtual AR monitors, and a hybrid combination of both for performing serious productivity work and found that virtual monitors are a feasible option, though currently constrained by lower resolution and field of view (see Figure 1.1). Such virtual AR workspace allows users to have an unlimited number of screens, screens of any size, and visual privacy. Users can also embed these virtual application windows in their surroundings (e.g., calendar and map on the wall, notepad on the table, email client at the right-hand side) to locate apps quickly and to prevent app windows from occluding important objects in the environment (see Figure 1.3(A)) [37][38]. To arrange 3D

planar windows in space, Lee et al. [39] proposed *Projective Windows* technique which strategically uses the absolute and apparent sizes of the window at various stages of the interaction to enable grabbing, moving, scaling, and releasing of the window in one continuous hand gesture. Instead of manually managing these application windows, Lindlbauer et al. [40] tried to automatically adapt which applications are displayed, how much information they show, and where they are placed based on users' current cognitive load, knowledge about their task, and environment (see Figure 1.2). Moreover, while users are leaving their desk, they can take their office with them, and interfaces adapt dynamically (e.g., register windows to the head/body/world, avoid occlusion by changing windows layout/transparency, switch the primary interaction technique, etc.) according to their walking and physical environment (see Figure 1.3(B & C)) [41][42][43]. Users can also continue on-the-go learning by watching the video in AR-HMDs. Ram and Zhao [44] studied different video presentation styles on AR-HMDs to distribute better user's attention between learning and walking. They introduced Layered Serial Visual Presentation (LSVP) style for future AR-HMD based on-the-go video learning. Further, Lu et al. [43] proposed *Glanceable AR*, an information access paradigm for AR-HMDs in walking scenarios. In this paradigm, applications (e.g., weather app, email, calendar, etc.) reside outside FOV to stay unobtrusive and can be accessed by a quick glance (e.g., eye-glance, head-glance, and gaze-summon) whenever needed. In this way, AR-HMDs provide us the unprecedented ability to visualize digital information anywhere seamlessly.

Figure 1.1: Comparing physical monitors, virtual AR monitors, and a hybrid combination of both for productivity tasks [36]: (a) Physical condition had three monitors side-by-side; (b) Virtual condition had three monitors rendered through HoloLens; (c) Hybrid condition combined a central physical monitor with two peripheral virtual monitors.

Particularly, for productivity work, these 2D virtual windows in AR-HMDs often contain textual information (e.g., pdf/-word document, excel sheet, or web browsing), and users need to perform text input and editing operations frequently

Figure 1.2: (A) activity: doodling, cognitive load: low; (B) activity: brainstorming, cognitive load: medium; (C) activity: reading scientific paper, cognitive load: high. Based on the current context, Mixed Reality interfaces adapt automatically [40]. For example, the interface shows more elements in more detail in the case of a low cognitive load task (A). Whereas increased cognitive load leads to a minimal UI with fewer elements at lower levels of detail (B to C).

Figure 1.3: (A) In an AR workspace, virtual windows are embedded into the user's nearby surfaces [38]; (B & C) AR interfaces adapt dynamically based on the user's walking and physical environment [41].

[2]. When they are at their desk, they can take advantage of a standard keyboard and mouse/trackpad to do these tasks efficiently. But, while users are in on-the-go situations, achieving text input and editing becomes non-trivial as there are no equivalent input techniques of keyboard and mouse/trackpad. Recently, text input for AR-HMDs has gained significant attention from the research community [3–6]. Whereas limited research focused on editing text that a user has already typed [7]. Normally, text editing is a complex task, and the first step is to select the text to edit it. Therefore, in this thesis, we will focus on this text selection part.

Beyond AR-HMDs, it is also possible to augment our physical space directly with digital content using projectors. This technique is known as spatial augmented reality (SAR). In SAR, we can use a single static or steerable and multiple projectors to increase the potential display area and enhance the image quality. For monoscopic projections, there is no need to track the user's viewpoint to render suitable graphics. Tracking might be required for view-dependent renderings in some applications, such as accurate lighting effects. To experience view-dependent stereoscopic projections, users need to wear light-weight 3D shutter glasses with attached

markers for 6 DOF tracking. However, it is still possible to achieve multiple perspective views without wearing any glasses in a dyadic SAR scenario [45], but it requires each user to stay opposite to one another for maintaining correct display registrations.

Unlike HMDs, displays are always detached from the users and integrated into the environments in SAR. This enables multiple users to view the same augmentation with natural depth cues (except view-dependent stereoscopic projection, usually a single-user setup). They can touch and interact directly with physical surfaces, which provides passive haptic feedback and greatly enhances their understanding of the presented information. In addition, they can freely move around the physical mock-up and experience the augmentations from different angles and positions. Users can read each other's facial expressions during collaborative tasks as their faces are not covered with HMDs anymore. Further, collaboration is not limited by the amount of equipment available, rather by the number of people who can be accommodated in the available space. Researchers already developed several shared SAR environments to support co-located collaboration. For instance, *Augmented Surfaces* [46] allows users to display and transfer digital information freely among portable computers, tables, and wall displays (see Figure 1.4(A)). Users can also attach digital data (e.g., editorial instruction voice note) to physical objects (e.g., a VCR tape) on the table. *DigitalDesk* [47][48] adds digital properties to the physical paper (e.g., users can point at a number printed on a paper to enter it into a calculator; this is shown in Figure 1.4(B)). Whereas *PaperWindows* [49] takes the physical affordances of a paper to manipulate digital content projected on it (e.g., flip a paper to navigate to the next page of the document). Figure 1.4(C) shows a working *PaperWindows* prototype. Laviole & Hachet [50] track a sheet of paper to project an image for creating physical drawings. Moreover, the *Office of the Future* [51] envisions to use all surfaces (walls, tables, floors, etc.) inside an office space as displays by replacing the normal office lights with projectors and cameras.

SAR is not only limited to planar surfaces. *Shader Lamps* [52] uses projectors to directly illuminate the surfaces of a physical model to alter its visual properties, such as colors, textures, lighting, shadows, etc. (see Figure 1.5). This table-top SAR has also been extended to the room-scale [53] [54]. *IllumiRoom*[53]

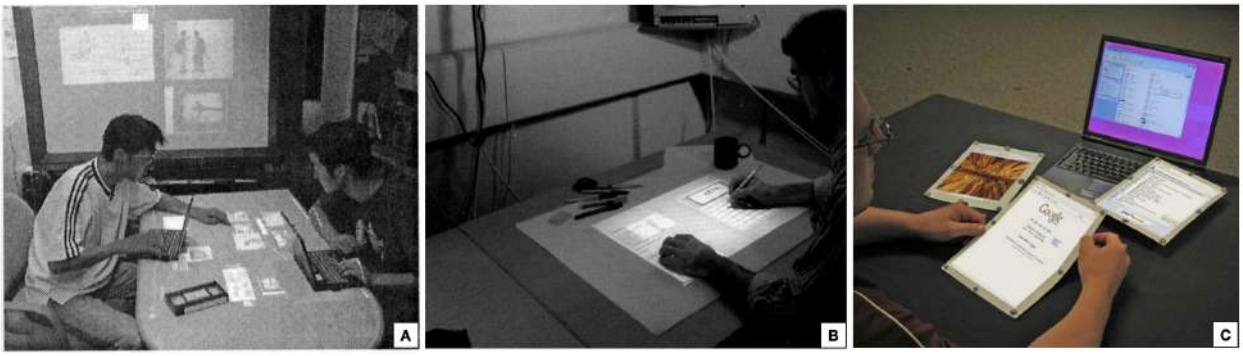

Figure 1.4: (A) Collaborating through *Augmented Surfaces* [46]; (B) *DigitalDesk* prototype where users selected a printed number on a piece of paper and put that number into the calculator application [47]; (C) *PaperWindows* prototype with three pages [49].

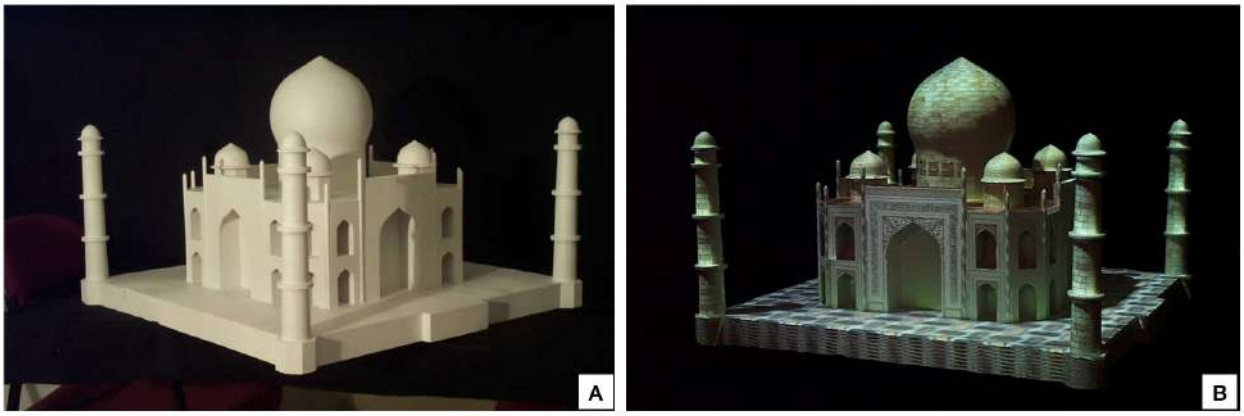

Figure 1.5: (A) The physical model of the *Taj Mahal*. (B) This model is illuminated with projectors to simulate different materials [52].

augments an environment around a conventional display with projected visualizations to expand the visual display area as well as to create a magical gaming experience (see Figure 1.6). *RoomAlive* [54] takes this concept to the next level to transform an entire room into an immersive augmented display and allow multiple users to interact with the projected content directly in a playful way. Furthermore, recent work focuses on augmenting deformable surfaces [55] and movable objects [56] instead of limiting SAR only to stationary surfaces. Note that, in this thesis, we consider a most commonly used collaborative SAR set up like *Shader Lamps* [52].

Compared to head-mounted AR display, SAR provides an unlimited field of view (theoretically), a scalable resolution, and an easier eye accommodation (because virtual objects are typically rendered on the physical surfaces). Despite all these advantages, bringing interactivity in the SAR environment is not straightforward. Users often need graphical widgets to interact with the SAR scene. Unfortunately, supporting UI

Figure 1.6: *IllumiRoom* is a proof-of-concept system that augments the physical environment surrounding a television to enhance interactive experiences [53]. (a) With a 3D scan of the physical environment, we can (b) directly extend the FOV of the game, (c) selectively render scene elements.

control elements in spatial augmented reality is challenging due to rigid mapping between the physical surface and virtual content. Therefore, this dissertation will look into how to embed widgets elements in the SAR scene.

1.1 Research Questions

Overall, this thesis aims to enrich the interaction space of augmented reality by addressing two main research questions, which we have described below.

In AR-HMDs, we can see a text document (e.g., a word/pdf file, web pages) in full size, but we can not select text efficiently. Performing text selection, which requires character level accuracy, in on-the-go situations is cumbersome because currently available input techniques of an AR-HMD (hand tracking, head/eye-gaze, and voice) are not as efficient as a standard keyboard and mouse/trackpad. For example, hand tracking often lacks precision [9] and is not suitable for long-term use [11]. Both eye-gaze and head-gaze have 'Midas Touch' problem (i.e., unintended target selection) [14]. Further, voice interaction is not socially acceptable in public places [17]. Therefore, in this work, we asked ourselves — *is there a better way (i.e., interaction technique) which will allow users to select text fast and accurately in an AR-HMD?*.

On the other hand, in spatial augmented reality, digital augmentation is always pre-defined, and the user often acts as a passive viewer. A way to enhance the interactivity in SAR environment is to make graphical widgets (pop-up windows, menus, labels, interactive tools, etc.) more accessible. Previous works attempted to provide widget elements in SAR on the surface of a table [31], on a tracked panel [57], or via a handheld tablet device [58]. Those approaches have

limitations. For instance, when UI elements are located on the table, users always have to come close to it to access widgets while viewing the scene from a distance. In other cases, their hands are occupied to carry a panel/tablet, and they have to divide their attention between augmented scenes and tools at their hands. By displaying widgets in mid-air like *floating menus* in AR-HMDs, we might overcome these issues. Then, physical augmentation and widgets will be in the user's view. Unfortunately, displaying virtual content in the air is not feasible in SAR as it always requires a physical surface for projection. Therefore, in this research, we asked ourselves — *how to provide contextual graphical widgets in mid-air in SAR?*

1.2 Research Contributions

The research contributions of this thesis are the following:

First, we developed four text selection techniques for AR-HMDs using a smartphone as an input device because of its availability, familiarity, social acceptability, and tangibility. All these techniques are eyes-free and one-handed. Our proposed four techniques are — continuous touch (where smartphone touchscreen acts as a trackpad pad), discrete touch (where smartphone touchscreen is used to move the cursor character by character, word by word, and line by line), spatial movement (smartphone is used as an air-mouse), and raycasting (smartphone is used as a laser pointer). Next, we compared these techniques in a user study where users have to select text at various granularity levels. Our results suggest that continuous touch outperformed the other three techniques in terms of task completion time, accuracy, and user preference.

Second, we proposed *DroneSAR* to provide interactive graphical widgets in SAR like a floating menu in mid-air using projection on a drone-mounted panel, while users issue eyes-free input via a hand-held controller to interact with the scene and positioning the drone dynamically. Drones can be positioned quickly with an acceptable accuracy around the augmented scene. As a result, users can freely move in the space, and they can access widgets when needed from anywhere in the scene. We present three possible ways to embed widgets using a drone in SAR — displaying annotations in mid-air, providing interactive tools, supporting different

viewpoints. We also describe the implementation details of our approach.

1.3 Organization of the Thesis

This dissertation is organized in the following way. Chapter 2 briefly covers the overall interaction space of AR-HMDs and SAR. Then, in Chapter 3, we describe our proposed smartphone-based text selection techniques in AR-HMDs. Next, Chapter 4 presents our proposed *DroneSAR* system to support contextual graphical widgets in mid-air in a SAR scene. Finally, Chapter 5 summarizes this thesis and provides future directions for further exploration.

1.4 Publications

All main contributions of this thesis have been published at international peer-reviewed conferences. Here are our publication details —

1. Rajkumar Darbar, Arnaud Prouzeau, Joan Odicio-Vilchez, Thibault Lainé, and Martin Hachet. "*Exploring Smartphone-enabled Text Selection in AR-HMD*". In:Proceedings of Graphics Interface. 2021. (GI '21). Virtual Event: Canadian InformationProcessing Society. pp. 117-126. doi: https://doi.org/10.20380/GI2021.14

2. Rajkumar Darbar, Joan Sol Roo, Thibault Lainé, and Martin Hachet. "*DroneSAR: Extending Physical Spaces in Spatial Augmented Reality using Projection on a Drone*". In:Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia. 2019. (MUM '19). pp. 1-7. doi: https://doi.org/10.1145/3365610.3365631

Related Work 2

In this chapter, we will first briefly review the overall interaction space of head-mounted augmented reality displays (AR-HMDs). Then, we will take a quick look into spatial augmented reality (SAR) interaction space.

2.1 Interaction Space in AR-HMDs

The overall interaction space of an AR-HMD consists of input techniques (hand-tracking, head-gaze, eye-gaze, body, pen/stylus, mobile devices, and multimodal) and extending the output (i.e., display). The following subsections briefly describe this space.

Hand Tracking Based Interaction

In current AR-HMDs, hands are the primary input modality as it allows natural user interaction (i.e., intuitive and easy to learn), thanks to RGB and depth cameras installed on the device. Using hands, users can manipulate 3D virtual objects as well as interact with 2D user interface elements (windows, virtual keyboard, menus, etc.) directly [59] or indirectly [60]. For example, Gloumeau et al. [61] proposed a novel 3D object manipulation technique, PinNPivot, which allows users to place a pin anywhere on the object to restrict its rotation during direct manipulation. The pin acts as a pivot for rotation. With this technique, a user locks an accurately placed part and continues interacting with the rest of the object without having to be concerned about displacing the locked part. Figure 2.1 illustrates this technique.

Arora et al. [62] investigated the use of hand gestures for authoring animations in AR/VR. In particular, they conducted an empirical study that explores user preferences of mid-air gestures for creating and editing dynamic, physical phenomena (e.g., particle systems, deformations, coupling) in animation. Figure 2.2 shows two example gestures from their study. Whereas, Piumsomboon et al. [63] developed a

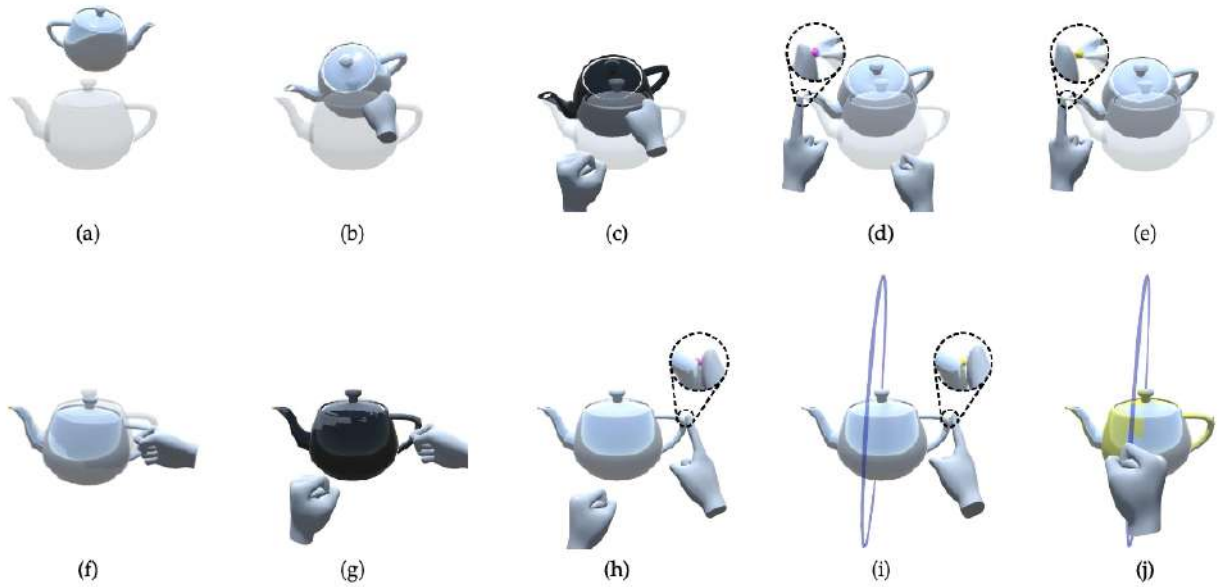

Figure 2.1: A docking task completed with PinNPivot [61]: (a) Source and target. (b) 6DOF manipulation quickly turns the object around. (c) Scaled 3DOF translation accurately places the spout's tip. (d) A pin is created. (e) It is locked. (f) The object is quickly rotated in 3DOF. (g) It is accurately rotated in scaled 2DOF. (h) A second pin is created. (i) It is locked, and a ring appears. (j) The object is rotated in scaled 1DOF. When the target turns yellow, it indicates a good fit.

comprehensive set of user-defined hand gestures for a range of tasks in augmented reality. VirtualGrasp [64] technique enables users to retrieve an object by performing a static barehanded gesture in mid-air (e.g., users can perform a gun-holding "hook" gesture to retrieve a virtual gun). Satriadi et al. [65] explored bare-hand gestures for performing panning and zooming operations to navigate the multi-scale map in AR. HandPoseMenu [66] offers to invoke the command with its corresponding hand-pose. Surale et al. [67] presented an empirical comparison of eleven bare hands, mid-air mode-switching techniques (including two baseline selection methods: bare hand pinch and device controller button) in mixed reality applications (see Figure 2.3). They found non-dominant hand techniques to be fast and accurate compared to most dominant hand techniques.

Researchers also explored the potential of hand tracking for tap and swype based text input [68] as well as ten finger mid-air typing [69][70] in mixed reality devices. While hand tracking, available in modern HMDs, allows for easy access to mid-air gestures, the accuracy of those spatial tracking solutions is still significantly lower than dedicated lab-based external tracking systems [71]. Moreover, it is not precise, suffers from "gorilla-arm" effects for extended periods of use,

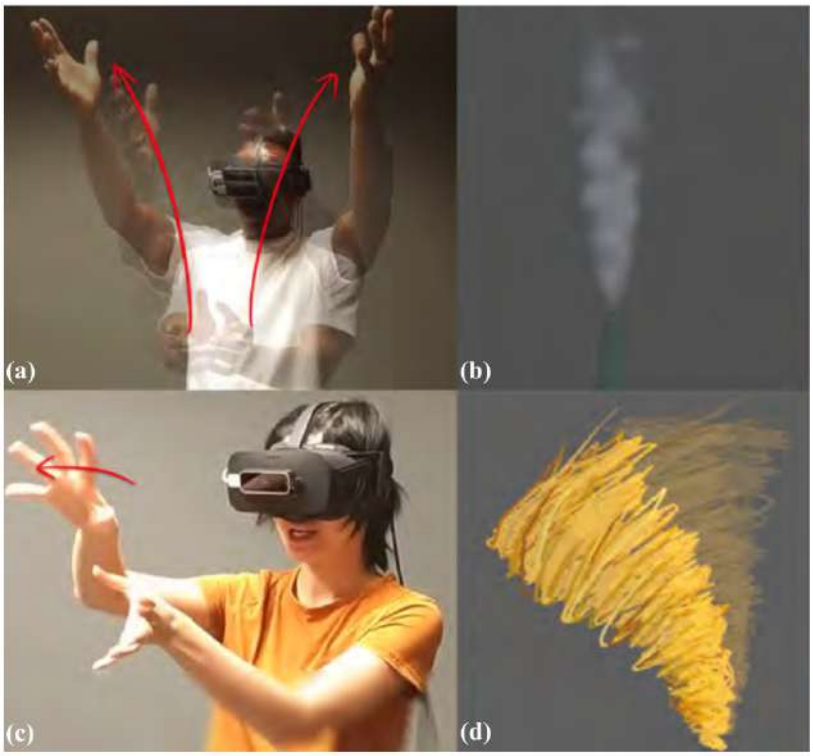

Figure 2.2: Two example gestures for authoring animations [62]: (a-b) a gesture manipulating the direction, spread, and randomness of smoke emission; (c-d) a gesture directly bending an object to describe a follow-through behaviour.

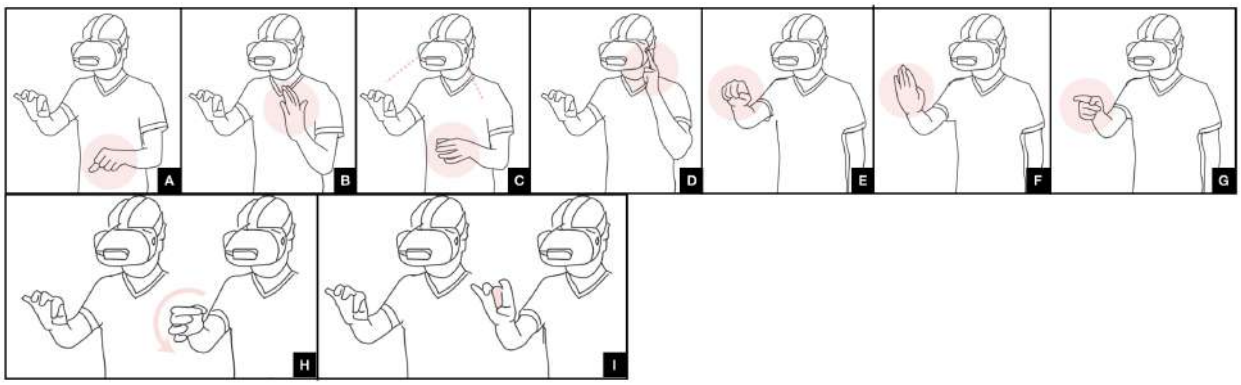

Figure 2.3: Selected mid-air mode-switching techniques [67]: (A) non dominant fist; (B) non dominant palm; (C) hand in field of view; (D) touch head; (E) dominant fist; (F) dominant palm; (G) point; (H) orientated pinch; (I) middle finger pinch.

and lacks tactile feedback. In comparison, touch interaction is accurate, tactile, familiar to users, and comfortable to use for prolonged usage. Xiao et al. [72] developed MRTouch to bring multitouch input capability in mixed reality by overlaying virtual content on the nearby physical surfaces (see Figure 2.4). Then users directly manipulate digital content affixed to that surface, which act as a virtual touchscreen.

Figure 2.4: MRTouch enables touch interaction in AR-HMDs [72]. (a) When a user approaches a surface, MRTouch detects the surface and (b) presents a virtual indicator. (c) The user touch-drags directly on the surface to (d) create a launcher and start an app. (e) In this app, the user uses touch to precisely rotate a 3D model.

Head-Gaze Based Interaction

Head movements are deliberate and accurate. To acquire a target with a head-gaze, users first position the cursor on the target with head movements, then confirm the target selection with another modality (e.g., button-press on a clicker, hand gesture, dwell timing). Yu et al. [73] first investigated the feasibility of head pointing based text typing for HMDs by proposing three techniques — DwellType (user selects a letter by pointing to it and dwelling over it for 400 ms), TapType (user selects a letter by pointing to it and tapping a button), and GestureType (user performs word-level input using a gesture typing style). They found that users can learn to type with their heads quickly, and it was not as fatiguing as it may seem at a first impression. Overall, GestureType outperformed the other two techniques.

In general, the confirmation techniques used in head-based pointing have their limitations. If the user's hands are busy, a button pressing on a clicker or performing a hand gesture might not be suitable. It is difficult to determine the optimal dwell time [74] as longer dwell time slows down the interaction, while shorter dwelling causes unintentional selection. Moreover, pre-defined dwell time makes interaction stressful because users must always be very focused and act carefully to avoid unwanted false selection. To overcome these challenges, Yu et al. [75] presented DepthMove, which allows users to interact with virtual objects proactively by making depth dimensional movements (i.e., moving the head perpendicular to the AR/VR-HMD forward or backward). Using this approach, Lu et al. [76] proposed a novel hands-free text entry technique, called DepthText (see Figure 2.5).

Inspired by our natural head movements (e.g., to nod or shake head to communicate), Yan et al. [77] developed a set of head gestures (see Figure 2.6) using users' intentional head

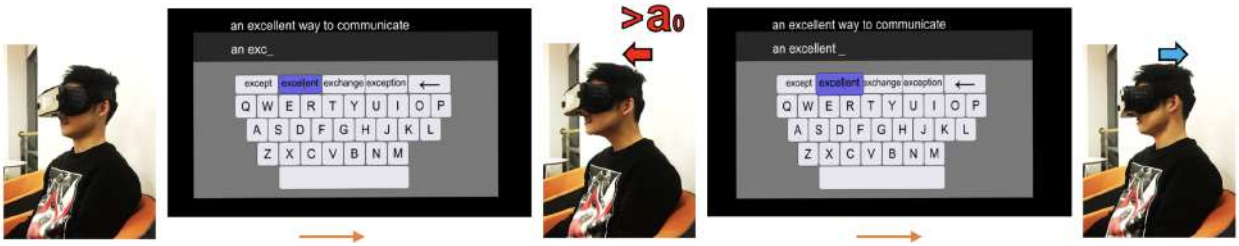

Figure 2.5: A user is entering text using DepthText in a VR-HMD. The user has entered the word “excellent" by performing short forward movements towards the depth dimension (z-axis) with an acceleration speed larger than threshold a_0

movements to support basic operations (such as pointing, dragging, zooming in and out, scrolling up and down, and returning to the homepage) in AR-HMDs.

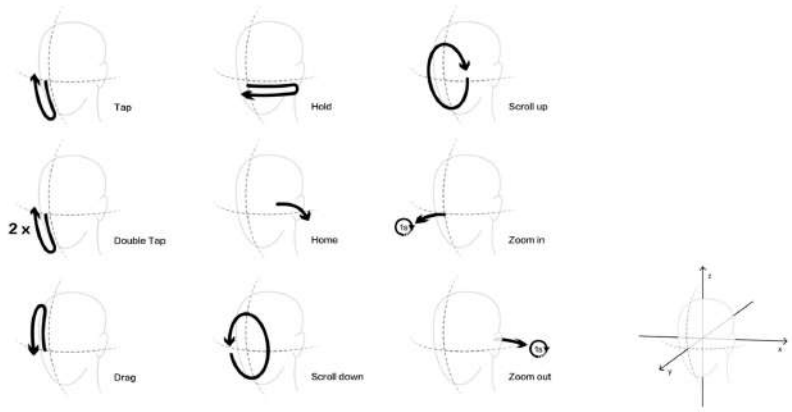

Figure 2.6: Head gestures set for different commands [77]. The movement of head is indicated by the arrows. “2×" represents the repeating of the action for twice. “1s" is an illustration for a dwell.

Yan et al. [15] proposed HeadCross for performing both selection and confirmation in one step using only head movements. With this technique, a user moves the pointer to approach the target and performs a HeadCross gesture to select the target and confirm the selection at the same time. The HeadCross gesture, which is inspired by crossing-based selection methods [78], requires users to move the pointer across the target boundary and then turn it back immediately. This “inside-outside" design rejects the false positives of unintentional head movements and speeds up the selection process. They showed several potential applications such as text input, menu selection in HMDs using this approach.

Eye-Gaze Based Interaction

Eye-gaze input is ergonomic in nature and allows fast pointing, but it often suffers from the well-known ‘Midas Touch’ problem [12] of involuntary selection. To address this issue,

researchers investigated different solutions mostly based on dwell time [79], smooth pursuits (where eye gaze follows a target continuously) [80], closing one eye [81], gaze gestures [82], and also by using a second modality for confirming selections (e.g., hand gesture or a button press as in HoloLens). Among all these target confirmation methods, dwell time is the most popular one as it is simple to understand and easy to implement. Considering its importance, Fernandez et al. [83] presented GazeWheel, a visual feedback technique to improve its usability and performance. In GazeWheel, a visual feedback widget in the shape of a wheel is filled as the user is temporally closer to the selection event. When completely filled, a selection is made where the user is gazing. Recently, Mutasim et al. [84] compared three target confirmation techniques (i.e., pinch hand gesture, dwell time, a button click) while selecting the target with eye-gaze pointing. Their Fitts' law task experiment did not find any significant differences in terms of execution time, error rate, and throughput among all these techniques. Still, participants preferred button click and dwell over pinch because pinch was sometimes frustrating due to recognition errors.

Prior work also looked into eye-gaze based 3D object selection and manipulation. For instance, Sidenmark et al. [85] proposed a novel technique called Outline Pursuits to support object selection in occluded environments with eye-gaze. This technique combines three concepts — cone-casting, outlining, and motion generation for object selection by smooth pursuit. Figure 2.7 illustrates how Outline Pursuits works. Liu et al. [86] implemented three methods to manipulate 3D object rotation using only eye-gaze — RotBar (it enables per-axis rotation; it maps a 360^o rotation to a bar for each axis), RotPlane (it makes use of orthogonal planes to achieve per-axis angular rotations), and RotBall (it combines a traditional arcball with an external ring to handle user-perspective roll manipulations). Figure 2.8 illustrates these three techniques. Their results showed that RotBar and RotPlane were faster and more accurate in performing single-axis rotations, but that RotBall greatly outperformed the other two methods for multi-axis rotations.

Eye-gaze cues (e.g., fixations, saccades, or blinks) play an important role (predicting another person's intention, complex social signal, etc.) to achieve successful face-to-face communication. Similarly, researchers conducted several studies to

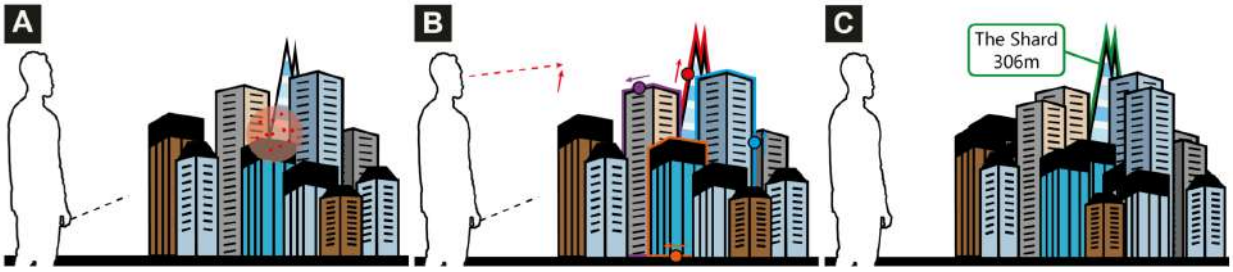

Figure 2.7: Outline Pursuits [85] support selection in occluded 3D scenes. A: The user points at an object of interest, but the selection is ambiguous due to occlusion by other objects. B: Potential targets are outlined, with each outline presenting a moving stimulus that the user can follow with their eye-gaze. C: Matching of the user's smooth pursuit eye movement completes the selection. Note that outline pursuits can augment manual pointing as shown or support hands-free input using the head or gaze for initial pointing.

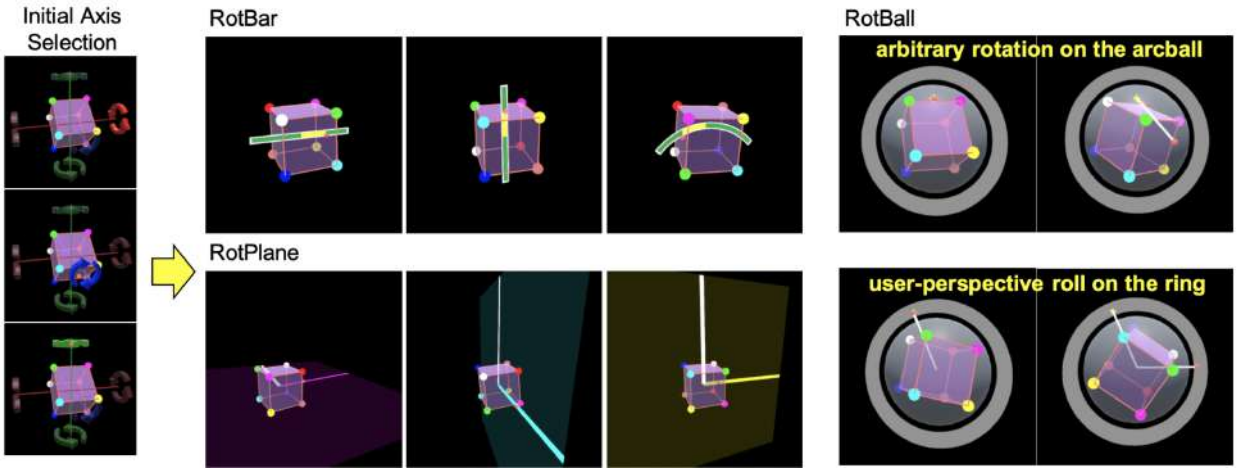

Figure 2.8: Three interaction methods proposed by Liu et al. [86] to manipulate 3D object rotation.

explore gaze visualizations in mixed reality remote collaboration [87][88][89]. But, they only considered uni-directional gaze visualization. Therefore, Jing et al. [90] prototyped the eyemR-Vis system (see Figure 2.9) to share dynamic gaze behavioural cues between a local host and a remote collaborator in a bi-directional way. This enables a more engaging and effective remote collaboration experience.

Body Based Interaction

In the past, researchers have utilized our body as an interaction surface to leverage our proprioception capabilities. This proprioception (i.e., the sensation of relative position, orientation, and movement of our body parts to each other) allows building user interfaces in an eyes-free way. Moreover, our body surface offers natural tactile cues when it is touched. With the proliferation of low-cost sensors and advancement

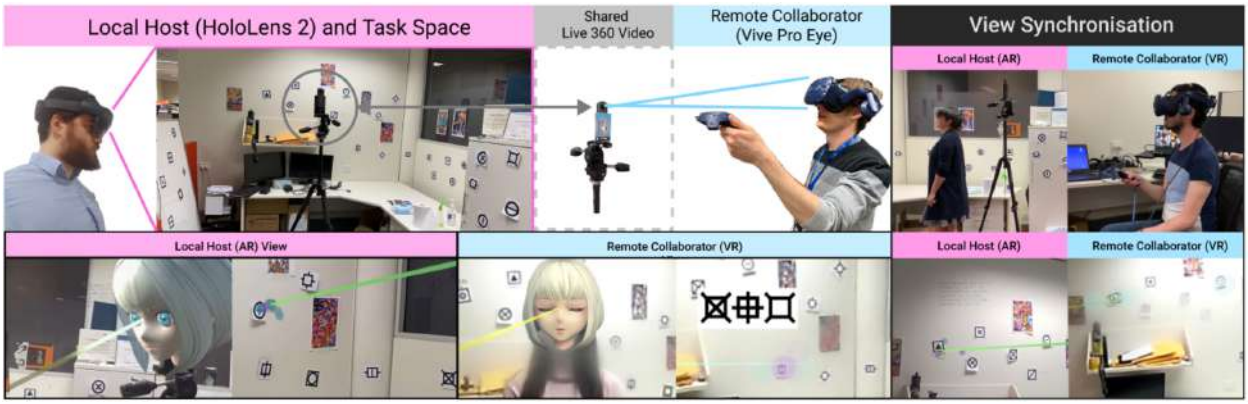

Figure 2.9: The eyemR-Vis prototype system, showing an AR user (HoloLens2) sharing gaze cues with a VR user (HTC Vive Pro Eye) and vice-versa.

in computer vision, prior work considered various parts of the human body as an input modality for interacting with mixed reality devices. For example, OmniTouch [91] projects virtual content on the user's body (forearm, palm, back of the hand) and allows them to directly manipulate it with multi-touch interaction. A depth camera was used to track fingers on the body. In PalmType, Wang et al. [92] mapped the QWERTY keyboard onto users' palms to enable text entry for HMDs (see Figure 2.10). This technique leverages users' natural ability to pinpoint specific areas of their palms and fingers without looking at them (i.e., proprioception) and provides visual feedback via wearable displays. Further, they developed PalmGesture [93] to draw stroke gestures on the palm to interact with HMDs (see Figure 2.11). In both prototypes, they used infrared imaging for finger tracking on the palm. Gustafson et al. [94] explored the possibility of palm-based imaginary user interfaces.

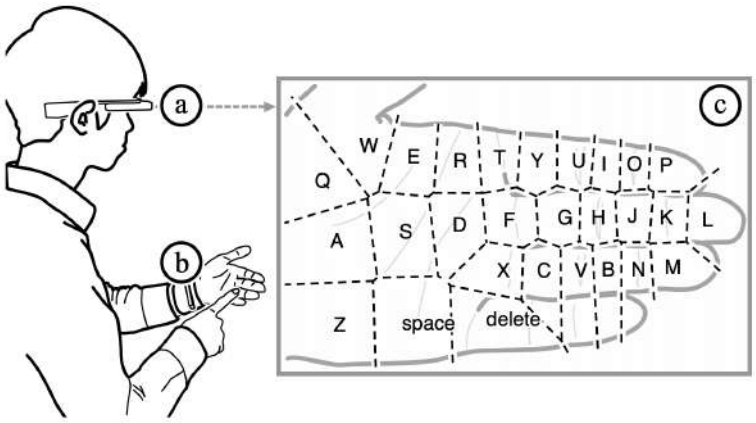

Figure 2.10: PalmType [92] enables text input for (a) smart glasses using a QWERTY keyboard interface, by using (b) wrist-worn sensors to detect the pointing finger's position and taps, and (c) displaying a virtual keyboard with highlighted keys via the display of the glasses.

Azai et al. [95] displayed a menu layout on the forearm, and users were directly interacting with it using touch, drag, slide, and rotation gestures (see Figure 2.12). In another work, they

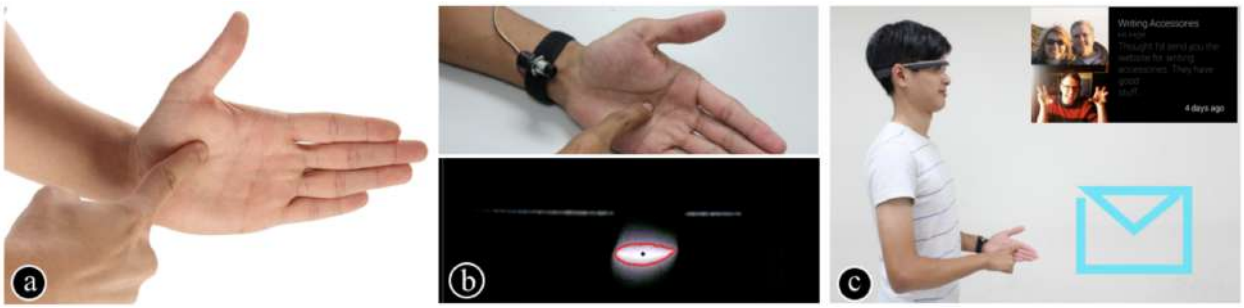

Figure 2.11: (a) Leveraging the palm as a gesture interface in PalmGesture [93]. (b) Tracking bright regions in the infrared camera image. (c) Drawing an email symbol to check emails on Google Glass.

proposed a Tap-Tap menu where hands, forearms, abdomen, and upper legs were used as menu display locations [96]. Researchers also investigated tap and word-gesture based subtle typing around the user's front thigh area (see Figure 2.13(A)) [97]. Muller et al. [98] systematically evaluated foot-taps as a direct and indirect input modality for AR glasses (see Figure 2.13(B)). In direct interaction, interfaces were displayed on the floor, and users were required to look down to select an item with foot tap. While in indirect interaction with users' feet, interfaces were displayed as a floating window in front of them. DMove [16] supports directional body motion-based interaction for AR display without requiring any external trackers (see Figure 2.13(C)). Lee et al. [99] extended previous implementations of single touch sensing nails [100] to all five nails and proposed an input space of 29 viable nail touches which includes taps, flicks, and swipes gestures. Our face (particularly cheeks area) has also been considered as an input surface to interact with HMDs [101]. Researchers studied different hand-to-face gestures (e.g., panning, pinch zooming, cyclo zooming, rotation zooming) for tasks involving continuous input such as document navigation. Recently, Weng et al. [102] developed a computer vision-based hand-to-face gesture sensing technique by fixing a downward-looking infrared camera onto the bridge of an AR glass. Other work explored the unique affordances of the human ear [103][104] and belly [105] for eyes-free interaction in HMDs.

Pen Based Interaction

On a desktop, we can easily point at small targets precisely with a mouse or trackpad. But, achieving a similar level of accuracy with current input techniques (hand-tracking, head/eye-gaze, voice) in AR-HMDs is cumbersome. Recently,

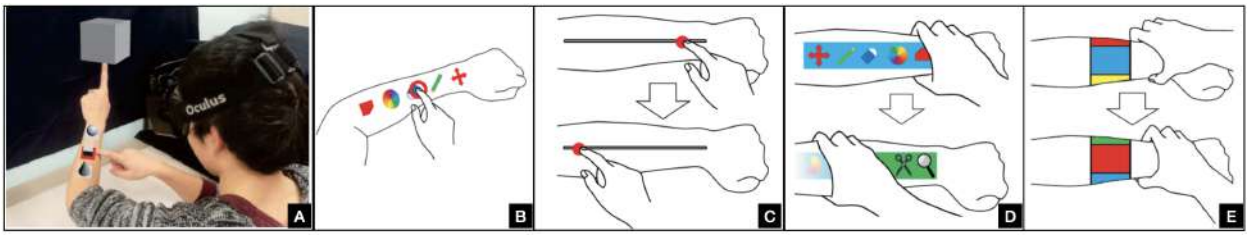

Figure 2.12: (A) The menu widget concept on forearm. A user is interacting with the menu using (B) touch (C) drag (D) slide and (E) rotate gestures.

Figure 2.13: (A) Word-gesture and tap based typing around the thigh [97]. (B) Foot-taps as a direct and indirect input modality for interacting with HMDs [98]. (C) A user needs to go the North-East direction using DMove technique and a selection is made when the user (nearly) completes the action [16].

several VR/AR pens (such as Massless*, Stylus XR†, VR Ink Stylus‡) have been introduced to support precise interaction with virtual content. Pham et al. [106] conducted a user study to compare three input devices (a mouse, a VR controller, and a 3D pen) on a VR and AR pointing task (see Figure 2.14). They found that the 3D pen significantly outperformed modern VR controllers in all evaluated measures, and its throughput is comparable to the mouse. Participants also liked the 3D pen more than the controller. Further in this direction, Batmaz et al. [107] studied Fitts' task performance when a pen-like input device is held in a power or a precision grip for interaction with objects at different depths (see Figure 2.15). Results favored precision grip irrespective of the distance of targets from the user.

Researches also investigated the use of pens for drawing and modeling in AR. Gasques et al. [108] developed a rapid interactive prototyping tool called PintAR (see Figure 2.16). With this tool, users first sketch digital content using a pen and tablet. Then, they rely on an AR-HMD to place their sketches in the real world and manipulate it directly. In SymbiosisSketch [109], the authors proposed a hybrid sketching system that combines drawing in mid-air (3D) and

* https://massless.io/
† https://holo-light.com/products/stylus-xr/
‡ https://www.logitech.com/en-roeu/promo/vr-ink.html

Figure 2.14: Pointing tasks in VR (top) and AR (bottom) using a mouse, a VR controller, and a 3D pen [106].

Figure 2.15: A participant is holding the pen with (A) a precision grip and (B) a power grip.

on a tablet device (2D) with a motion-tracked stylus to create detailed 3D designs of arbitrary scale in an AR setting.

Figure 2.16: PintAR [108] in use. (A) Designer sketching an interface element on the tablet using a pen. (B) Designer placing sketched element in the environment using an Air tap. (C) AR interface element placed side-by-side with real display.

Using AR, we can sketch directly on an existing physical object. This is particularly helpful in personal fabrication for novice designers. They can quickly and easily create simple 3D models that should attach to or align with existing objects (e.g., a snap-on handle for a drinks container). Sketching

directly on physical objects always provides natural haptic feedback whenever the pen touches them. Wacker et al. [110] studied the users' performance of a stroke when drawing with a pen around physical vs. virtual objects in AR (see Figure 2.17). They found that tracing physical objects was 48% more accurate but took longer than tracing virtual objects.

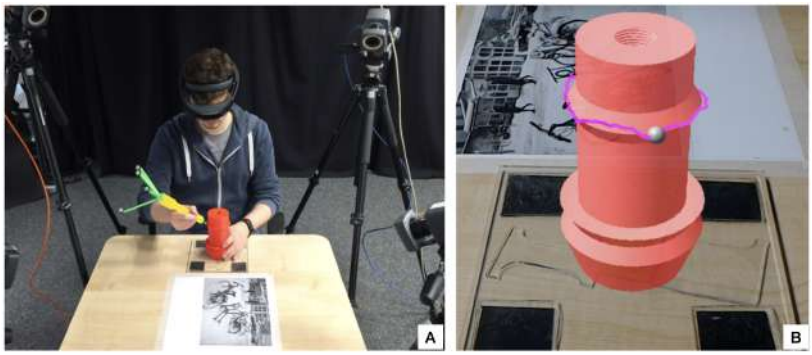

Figure 2.17: Participants were drawing a stroke around (A) a physical object and (B) a virtual object [110].

Beyond pointing and sketching, researchers explored pen-based text input techniques for AR/VR. Jackson et al. [111] designed two interfaces — Tilt-Type and Arc-Type. In Tilt-Type, the user first rotates the stylus left/right relative to his/her body to change the selected bin of characters. Then pitching the stylus toward/away from the body changes the selection within a bin (see Figure 2.18(B)). In Arc-Type, users first rotate the stylus around the z-axis by rotating their wrist to change the selected bin in the radial layout as shown in Figure 2.18(C). Then sliding the index finger on the stylus touch sensor changes the character selection within a bin.

Figure 2.18: (A) The overall setup of a stylus-based text input system in a CAVE environment [111]. (B) The Tilt-Type interface. (C) The Arc-Type interface.

Mobile Devices Based Interaction

By combining handheld devices and HMDs, researchers try to make the most of the benefits of both [20]. The handheld device brings a 2D high-resolution display that provides a multi-touch, tangible, and familiar interactive surface.

Whereas, HMDs provide a spatialized, 3D, and almost infinite workspace. With MultiFi [21], Grubert et al. showed that such a combination is more efficient than a single device for pointing and searching tasks (see Figure 2.19). For a similar setup, Zhu and Grossman proposed a design space for cross-device interaction between HMD and phone (see Figure 2.20). They demonstrated how it could be used to manipulate 3D objects [20]. Similarly, Ren et al. [23] demonstrated how it can be used to perform windows management (see Figure 2.21(A)). In VESAD [112], Normand et al. used AR to directly extend the smartphone display (see Figure 2.21(B)). Other work combined AR-HMDs and tablets for immersive visual data analysis (see Figure 2.22) [113][114].

Figure 2.19: MultiFi [21] widgets crossing device boundaries based on proxemics dimensions (left), e.g., middle: ring menu on a smartwatch (SW) with head-mounted display (HMD) or right: soft keyboard with full-screen input area on a handheld device and HMD.

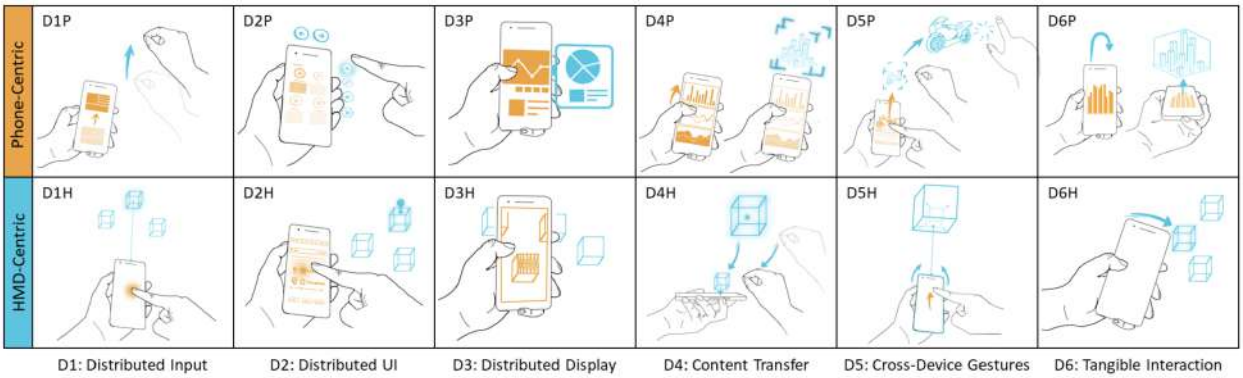

Figure 2.20: The BISHARE [20] design space of joint interactions between a smartphone and augmented reality head-mounted display. Each cell contains a single example of joint interaction, but represents a broader class of interaction techniques that may be possible.

Regarding the type of input provided by the handheld device, it is possible to only focus on using touch interactions, as it is proposed in Input Forager [115] and Dual-MR [116]. Waldow et al. compared the use of touch with gaze and mid-air gestures to perform 3D object manipulation and showed that touch was more efficient [117]. It is also possible to track the handheld device in space and allow for 3D spatial interactions.

Figure 2.21: (A) Understanding window management interactions using an AR-HMD + smartphone interface [23]. (B) Enlarging smartphone display with an AR-HMD [112].

It has been done in DualCAD, in which Millette and McGuffin used a smartphone tracked in space to create and manipulate shapes using both spatial interactions and touch gestures [22]. With ARPointer [118], Ro et al. proposed a similar system and showed it led to better performance for object manipulation than a mouse and keyboard and a combination of gaze and mid-air gestures. When comparing the use of touch and spatial interaction with a smartphone, Budhiraja et al. showed that touch was preferred by participants for a pointing task [119], but Büschel et al. [19] showed that spatial interaction was more efficient and preferred for a navigation task in 3D which is shown in Figure 2.23. In both cases, Chen et al. showed that the interaction should be viewport-based and not world-based [120].

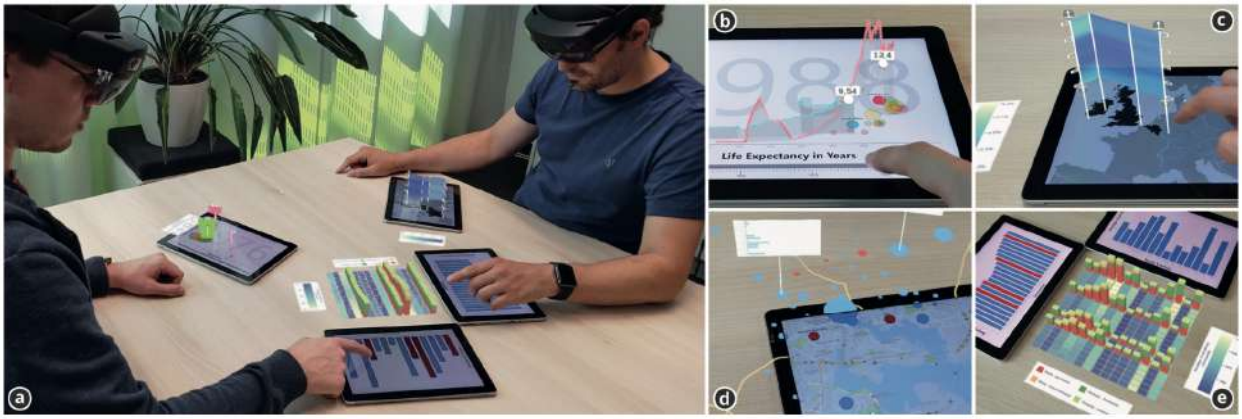

Figure 2.22: Data visualization using mobile devices and Augmented Reality head-mounted displays [113]: (a) Envisioned usage scenario; (b) 2D scatter-plot extended with superimposed 3D trajectories/paths; (c) 3D wall visualization in AR aligned with the mobile device; (d) Use of AR for seamless display extension around a geographic map; (e) Combining visualizations with an AR view between the devices.

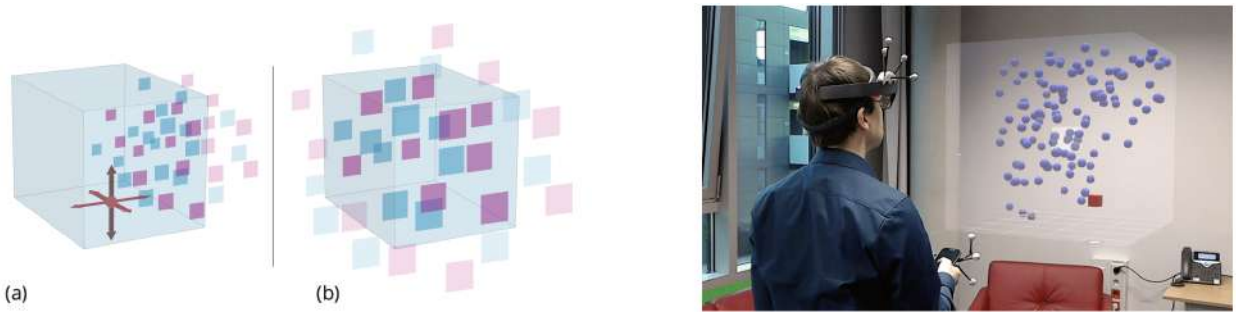

Figure 2.23: Left: 3D data spaces can be explored by (a) 3D panning and (b) zooming relative to their fixed presentation space. Right: A user wearing a HoloLens explores such a 3D data space with smartphone-based proposed interaction techniques [19].

Multimodal Interaction

Researchers explored rich interaction opportunities by combining multiple input channels. Complementing the strengths of multiple channels can lead to enriched user experiences.

Regarding multimodal inputs, the synergy between speech and gestures is a widely used input combination. When used as unimodal input, speech can be beneficial for abstract tasks, whereas gestures can be beneficial for direct pointing and manipulation. In 1980, Bolt [121] introduced a seminal work put-that-there interface which allows users to place objects inside a media room through a combination of speech and pointing gestures. Later this combination has been used in virtual object selection and manipulation in AR [122][123][124]. Piumsomboon et al. [125] studied the use of gestures and speech vs. only gestures for manipulating 3D objects in AR. Their results showed that gesture-only interaction outperformed multimodal technique substantially for most tasks (except object scaling). This indicates that multimodality per se is not always beneficial for interaction, but needs to be carefully designed to suit the task at hand. Further, Chen et al. [126] empirically investigated a set of techniques for disambiguating the effect of freehand interactions while manipulating virtual objects. They compared three input modalities (speech, head-gaze, and foot tap) paired with three different timings (before, during, and after an interaction) in which options become available to resolve ambiguity. The results indicated that using head-gaze for disambiguation during an interaction with the object achieved the best performance followed by speech and foot-tap. Prior work also looked into entering text using speech and hand-tracking [127] which is illustrated in Figure 2.24.

Figure 2.24: Typing on a midair auto-correcting keyboard with word predictions (left) vs. speaking a sentence and then correcting any speech recognition errors (right) [127]. Users correct errors by selecting word alternatives proposed by the speech recognizer or by typing on the virtual keyboard.

In recent times, researchers are trying to seamlessly combine eye-gaze with other modalities (such as mid-air gestures and head movements) for designing novel interfaces in AR/VR-HMDs. For example, Pfeuffer et al. [128] described the Gaze + Pinch technique which integrates eye-gaze with hand gestures for performing several tasks such as object selection, manipulation, scene navigation, menu interaction, and image zooming (see Figure 2.25).

Figure 2.25: Gaze + Pinch interactions unify a user's eye gaze and hand input: look at the target, and manipulate it (a); virtual reality users can utilise free hand direct manipulation (b) to virtual objects at a distance in intuitive and fluid ways (c).

In another work, Feng et al. [129] developed HGaze Typing, a novel dwell-free text entry system, which combines eye-gaze paths with head gestures. HGaze Typing allows explicit activation of common text entry commands (such as selection, deletion, and revision) by using head gestures (nodding, shaking, and tilting) and uses eye-gaze path information to compute candidate words. Their user study confirmed that HGaze Typing is robust to unintended selections and outperforms a dwell-time-based keyboard in terms of efficacy and user satisfaction. Kytö et al. [14] investigated both eye gaze and head pointing in AR-HMDs combined with a refinement provided by hand gesture input, a handheld

device, and scaled head motion. Their user study showed trade-offs of different multimodal techniques for precise target selection. Overall, head pointing was slower than eye-gaze input, but it allows greater targeting accuracy. The scaled head refinement proved to be the most accurate, although participants primarily preferred device input and found gestures required the most effort. Lastly, Radi-Eye, proposed by Sidenmark et al. [130], is a novel pop-up radial interface designed to maximize users' control and expressiveness with eye and head gaze inputs (see Figure 2.26). Radi-Eye provides widgets of both types — discrete (i.e., buttons) and continuous (i.e., a slider). Widgets can be selected with Look & Cross interaction [131] where eye-gaze is used for pre-selection followed by a head-crossing for confirmation.

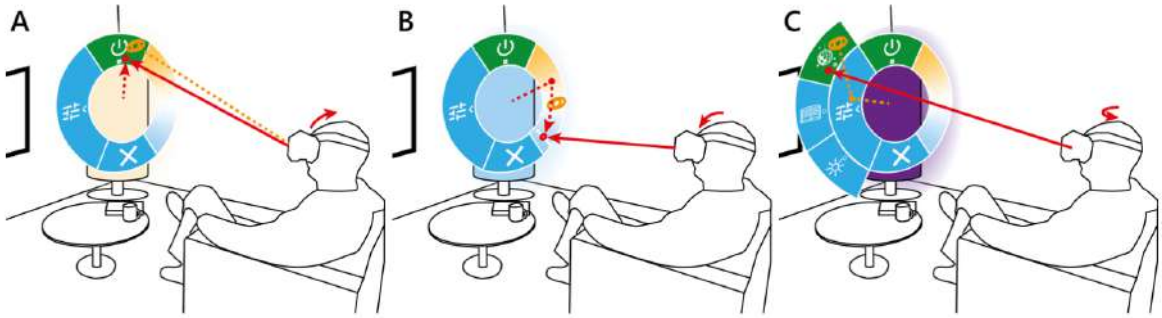

Figure 2.26: Radi-Eye [130] in a smart home environment for control of appliances. A: The user turns on the lamp via a toggle selection with minimal effort using only gaze (orange) and head (red) movements. B: Selection can be expanded to subsequent head-controlled continuous interaction to adjust the light colour via a slider. C: Gaze-triggered nested levels support a large number of widgets and easy selection of one of the multiple preset lighting modes. The widgets enabled via Radi-Eye allow a high level of hands-free and at-a-distance control of objects from any position.

Extending Display Space

While AR displays are effective in many ways by augmenting the physical world with digital information, it has a limited field of view (FOV), and the isolated user experience (i.e., virtual environment is only visible to the HMD user) makes collaboration with external users difficult. Previous research combined an AR-HMD with spatial augmented reality (SAR) [132], mobile devices (e.g., smartphone [21], tablet [133][113]), interactive surfaces [134], and large displays (see Figure 2.27) [135] to extend its display space and support external collaboration. Recently, AR-HMDs have also been combined with an actuated head-mounted projector to share AR content with co-located non-HMD users and enable them to interact

with the HMD user and become part of the AR experience (see Figure 2.28) [136][137][138].

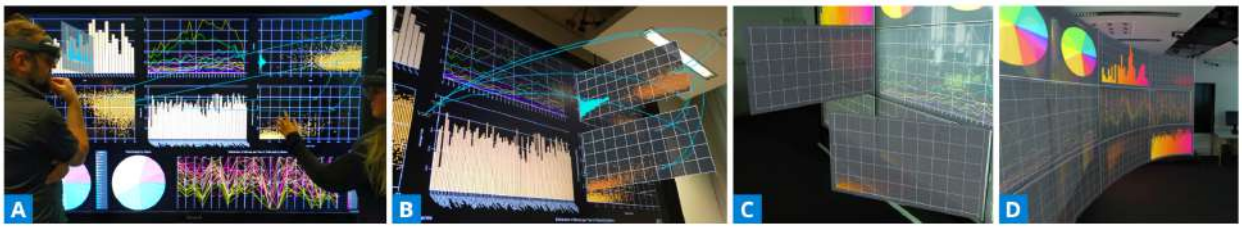

Figure 2.27: Extending large interactive display visualizations with Augmented Reality [135]. (A) Two analysts are working on data visualization tasks. (B) Displaying AR Brushing and Linking, Embedded AR Visualizations, and Extended Axis Views. (C) Hinged Visualizations to improve the perception of remote content. (D) Curved AR Screen is providing an overview of the entire display.

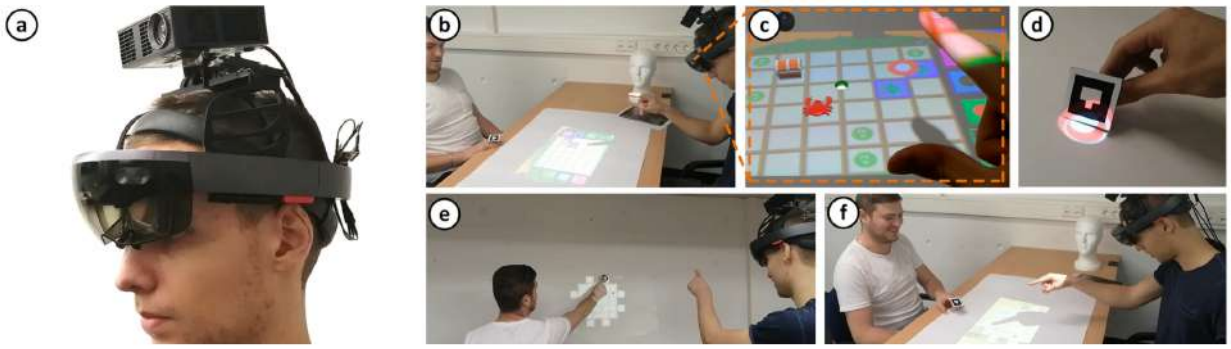

Figure 2.28: ShARe [137] is a modified AR-HMD consisting of a projector and a servo motor attached to its top (a). This allows people in the surrounding to perceive the digital content through projection on a table (b, f) or on a wall (e) and interact via finger-based gestures (c) or marker-based touch (d).

Summary of AR-HMDs Interaction Space: Previous work has proposed several interesting ways to enrich the interaction space of AR-HMDs. Although we have discussed both input and output space briefly, we will mainly focus on the input interaction for our purpose in this thesis. Researchers have tried to accomplish different tasks (such as object selection and manipulation, interactive mid-air sketching, menu selection, mode switching, text input, gaze cues visualization in remote collaboration, windows management, interactive data visualization and navigation, etc.) with different input techniques. But, the text selection task in head-mounted AR glasses didn't receive much attention from the community so far (note that we have mentioned those work that considered text selection in AR-HMDs in the specific related work section of Chapter 3). In this dissertation, we are particularly interested to investigate smartphone as an input device for selecting text in AR glasses. Overall, previous research showed smartphone provides a good alternative input for augmented reality display in various tasks. Still, it is not clear

for text selection — only tactile interactions should be used on the smartphone, or it should also be tracked in mid-air to provide spatial interactions. We have tried to address this question in this thesis.

2.2 Interaction Space in SAR

An advantage of spatial augmented reality (SAR) over AR-HMDs is that users can see the virtual content directly without wearing head-mounted displays. It enables multi-user collaboration seamlessly. However, the best way to interact with SAR setup is still an open problem.

To bring interactivity in the SAR scene, Bandyopadhyay et al. [31] initially extended the concept of shader lamps [52] with dynamic tracking to allow the users to paint onto physical objects using a tracked brush (see Figure 2.29).

Figure 2.29: Dynamic shader lamps for applying virtual paint and textures to real objects simply by direct physical manipulation of the object and a “paint brush" stylus. [31].

Hoffman et al. studied that the ability to touch physical objects significantly increases the realism of virtual experiences [139]. Ware and Rose [140] found that users’ ability to perform virtual object manipulations (such as rotating a virtual object) improves with physical handles. Although these experiments were conducted in the context of virtual reality, the results are applicable to the SAR environment undoubtedly. As users’ hands are free (they don’t need to hold a display in their hands) and they can see the real world in SAR, they can take advantage of physical objects as a part of the user interface elements like tangible user interfaces (TUIs) [141]. These objects can also act as display surfaces. Inspired by this, researchers tried to combined TUI and SAR together. For example, Jones et al. [142] developed a novel surface interaction engine that enables users to build their own

physical world with almost any material suitable for the projection (like a set of wooden blocks), map virtual content onto their physical construction, and play directly with the surface using a stylus (see Figure 2.30). Their setup provides end-users a uniquely immersive, tangible SAR experience. They also designed a set of surface adaptive 2D GUIs (like a radial menu) to show possible avenues for structuring interaction on complex physical surfaces.

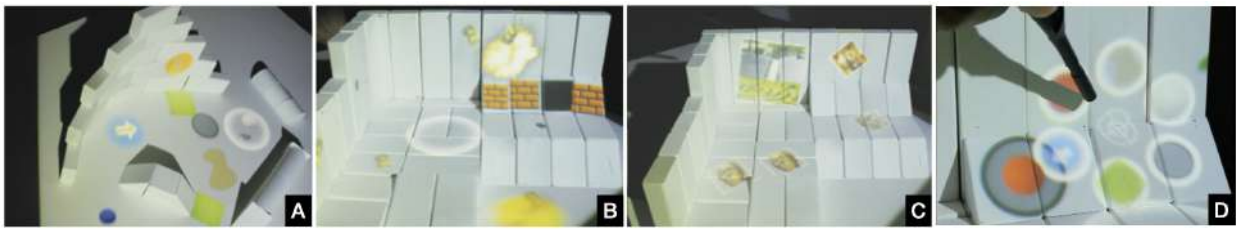

Figure 2.30: Motivating surface interaction examples of the “Build, Map, Play" process proposed by Jones et al. [142]: (A) a virtual miniature golf game, (B) a two-player tank game, (C) a photo viewer. (D) The user selects a menu item with a stylus from a surface adaptive radial menu that adapts to the surface that it is displayed on.

Marner et al. [57] proposed the concept of Physical-Virtual Tools (PVTs) in SAR by projecting application-specific GUIs on a physical tool that is carried by the user in the projection area. This allows to overload a single tool with several functions to interact with SAR. Figure 2.31) shows an example of PVTs for an airbrushing application. The user holds the stencil tool in his left hand, and the system projects the controls widgets on it. The airbrush tool, held in the right hand, is augmented with projection to provide currently selected paint color, spray angle, and brush type information. PVTs, build on the physical nature of TUIs, are invaluable in a large-scale SAR system where there is no fixed location available for displaying traditional user interfaces (UIs). Moreover, users can interact with the system using this tool from anywhere in the scene as they always have quick access to control widgets.

We can also utilize unique physical affordances of everyday objects (e.g., markers, stapler, coffee-mugs, etc.) beyond a dedicated tangible system to enhance interaction with the SAR system. For instance, users immediately pick up a marker from their desk and move/rotate in different directions to use it as an improvised joystick to control 3D content. Henderson and Feiner called this type of ad hoc interaction as “opportunistic controls" [143]. With inexpensive sensing technology like Microsoft Kinect[§], users can create such low-

[§] https://azure.microsoft.com/en-us/services/kinect-dk/

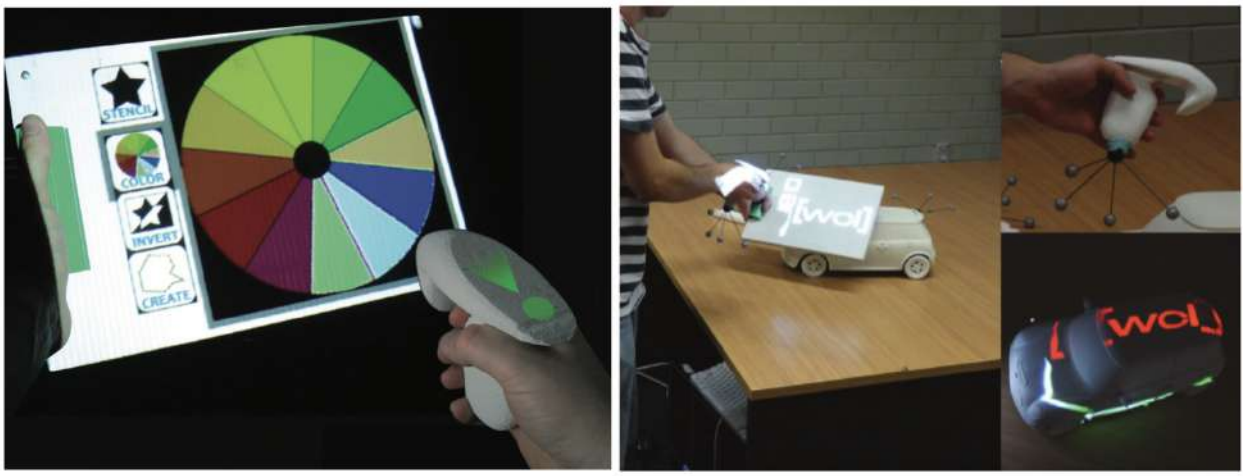

Figure 2.31: Augmented handheld tool is providing virtual widgets for a paint application [57].

fidelity input devices on the fly in an untethered way. Figure 2.32 illustrates two examples of ad hoc interactions.

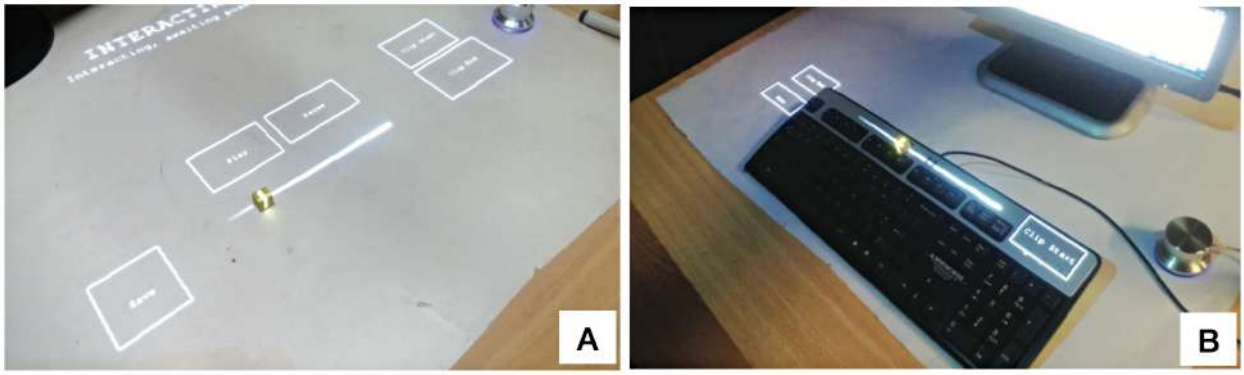

Figure 2.32: Examples of ad hoc controls, in which the wooden cube is the handle for slider interaction [144]. (A) Video-editing controls on their own. (B) The same controls supplementing the existing standard controls.

On the other side, Schmidt et al. [145] presented a floor-based user interface (UI) that allows multiple users to explore a SAR environment with both monoscopic and stereoscopic projections (see Figure 2.33). MirageTable [146] combined a depth camera, a curved screen, and a stereoscopic projector to enable the user to perform freehand interaction with virtual 3D objects in a tabletop SAR scenario (see Figure 2.34).

Other techniques explored mobile devices and a standard mouse to interact with the SAR scene. For example, Hartmann and Vogel [147] investigated three smartphone-based pointing techniques to select virtual objects in SAR (see Figure 2.35). These methods are — viewport pointing (use the phone's camera as a viewport to interact with remote targets), raycast pointing (a virtual ray emits from the phone's front end towards the target and then the user taps a button on the

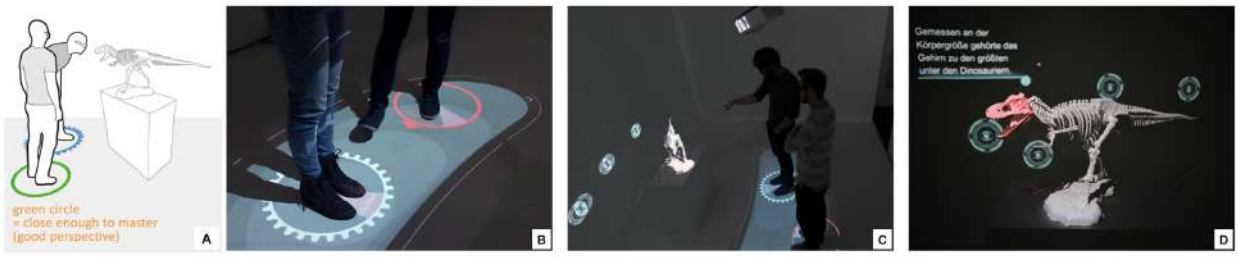

Figure 2.33: (A) The concept of floor projected UI in a collaborative SAR environment [145]. (B) Experimental setup with the extended floor UI. (C) Participants discussing a virtual scene. (D) The virtual scene from the master's (the user who controls the perspective) point of view.

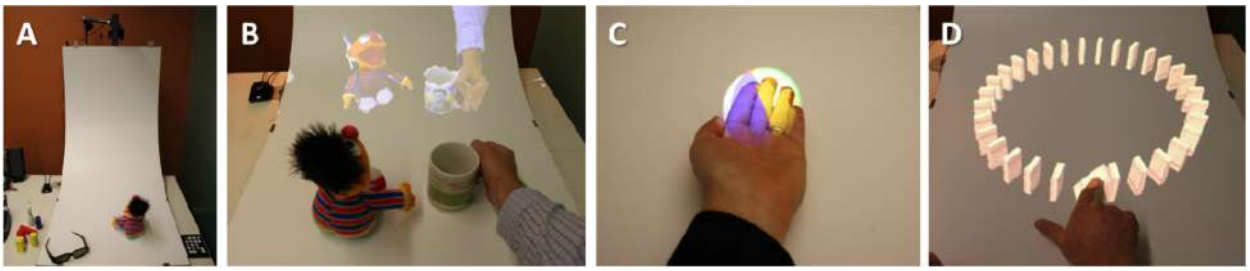

Figure 2.34: MirageTable is a curved projection-based augmented reality system (A), which digitizes any object on the surface (B), presenting correct perspective views accounting for real objects (C) and supporting freehand physics-based interactions (D).

screen to make a selection), and tangible pointing (a target is selected by directly touching it with the mobile phone). From the experiments, the authors found that raycast is fastest for high and distant targets, tangible is fastest for targets in close proximity to the user, and viewport performance is in between. Similarly, Park et al. [58] integrated mobile devices in projection-based AR to afford user interfaces to design interiors effectively. It is described in Figure 2.36(B). Gervais et al. [148] found that a standard mouse can be a good fit for pointing in a desktop SAR setup (see Figure 2.36(A)).

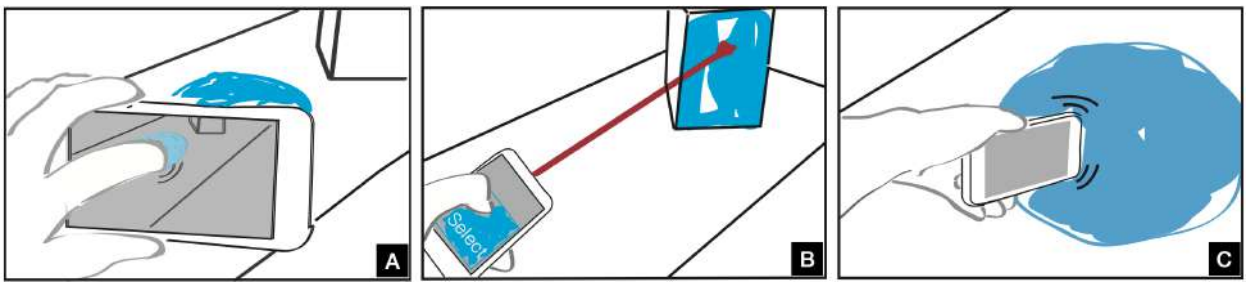

Figure 2.35: Mobile phone pointing techniques in SAR [147]: (A) viewport, where targets are captured by a camera-like view; (B) raycasting, where targets are pointed at; (C) tangible, where targets are directly contacted.

Due to the rigid mapping between physical and virtual parts in spatial augmented reality, the virtual scene cannot be explored in different scales and points of view. To overcome this issue, previous works fused multiple mixed reality

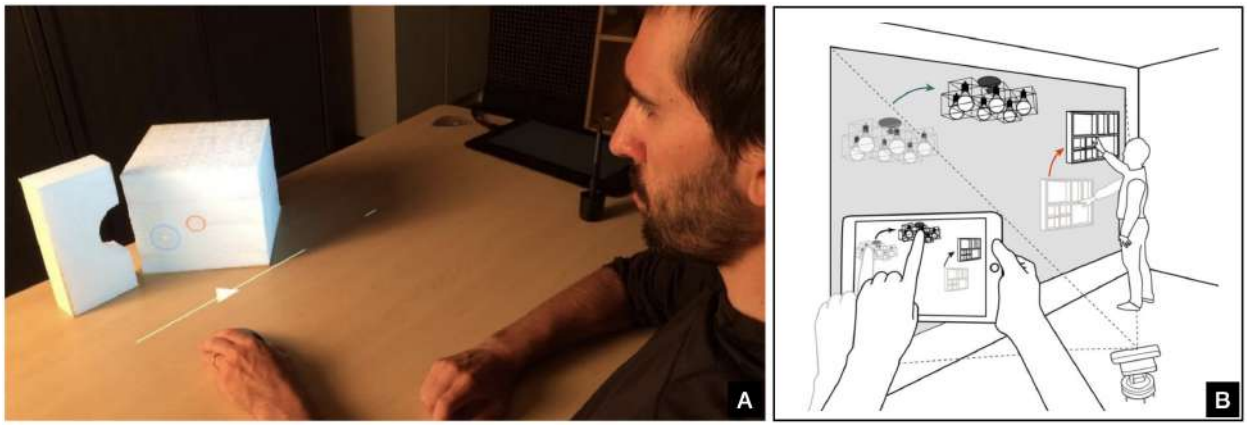

Figure 2.36: (A) The user is moving a cursor (represented in blue) to a target (represented in red) on an augmented object using a standard mouse [148]. (B) In DesignAR system, a user is manipulating projected content for interior design using a tablet [58].

modalities (like VR-HMD, hand-held see-through display) [33, 149]. In the One Reality framework [33], the authors first extended the SAR space with a see-through display by showing mid-air information (see Figure 2.37(middle)). Then, for further extension, users put HMDs, which provide a virtual replica of the physical scene, taking advantage of the freedom of virtual spaces without losing connection with the environment (see Figure 2.37(right)).

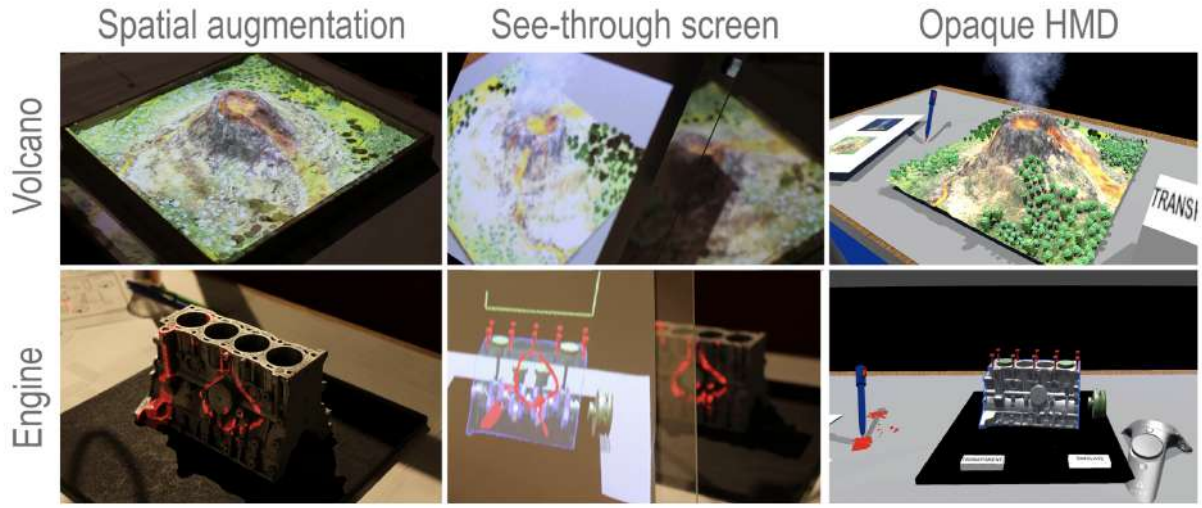

Figure 2.37: Example scenes to describe One Reality framework [33]: volcano mock-up made out of sand (top), 3D printed Toyota engine (bottom). Each scene can be interacted with different display technologies: spatial augmentation (left), see-through displays (middle), and opaque head mounted displays (right).

Summary of SAR Interaction Space:

Previous research proposed several interesting ways to interact with the SAR scene, but they have limitations too. Techniques that provide graphical widgets on the table (like

dynamic shader lamps [31], opportunistic controls interface [143][144]) or on the physical surfaces of the mock-ups [142] restrict users to be always close to the scene. This restriction might not be suitable in some scenarios like museums. Hand-based interaction [146] in SAR also has the same problem. With a standard mouse [148] indirect interaction is possible, but a surface is required to keep it (this limits our mobility in space). In the case of hand-held tool [57], users need to hold it all the time, as well as explicit attention switching is required to select UI elements projected on that panel. Floor projected UI and mobile devices [147][58] also have the focus shifting issue [145]. On the other hand, using other displays (VR-HMDs or transparent screens) [33, 149] to experience physical augmentation from different perspectives inhibits multi-user face-to-face collaborations in SAR.

To overcome these limitations, we proposed to provide interactive graphical widgets in mid-air using projection on a drone (this gives a sense of extending the space in SAR), while users issues eyes-free input via a hand-held controller (see Chapter 4). This combination of input and output allows users to maintain face-to-face collaborations; they can move in space freely and access interactive widgets instantly from anywhere without explicitly switching their attention.

Exploring Smartphone-enabled Text Selection in AR-HMDs

3

Chapter Summary: Text editing is essential and at the core of most complex tasks, like writing an email or browsing the web. Efficient and sophisticated techniques exist on desktops and touch devices, but are still under-explored for AR glasses. Performing text selection, a necessary step before text editing, in AR display commonly uses techniques such as hand-tracking, voice commands, eye/head-gaze, which are cumbersome and lack precision. In this work, we explored the use of a smartphone as an input device to support text selection in AR-HMDs because of its availability, familiarity, and social acceptability. We proposed four eyes-free text selection techniques, all using a smartphone — continuous touch, discrete touch, spatial movement, and raycasting. We compared them in a user study where users have to select text at various granularity levels. Our results suggested that continuous touch, in which we used the smartphone as a trackpad, outperformed the other three techniques in terms of task completion time, accuracy, and user preference.

3.1 Introduction

Text input and text editing represent a significant portion of our everyday digital tasks. We need it when we browse the web, write emails, or just when we type a password. Because of this ubiquity, it has been the focus of research on most of the platforms we are using daily, like desktops, tablets, and mobile phones. The recent focus of the industry on AR-HMDs, with the development of devices like the Microsoft HoloLens* and Magic Leap†, made them more and more accessible to us, and their usage is envisioned in our future everyday life. The lack of a physical keyboard and mouse/trackpad with such devices makes text input difficult and an important challenge in AR research. While text input for AR-HMDs has been already well-studied [3–6], limited research focused on editing text that a user has

* https://www.microsoft.com/en-us/hololens
† https://www.magicleap.com/en-us

already typed. Normally, text editing is a complex task and the first step is to select the text to edit it. This work will only focus on this text selection part. Such tasks have already been studied on desktop [150] with various modalities (like gaze+gesture [151], gaze with keyboard [152]) as well as touch interfaces [153]. On the other hand, no formal experiments were conducted in AR-HMDs contexts.

Generally, text selection in AR-HMDs can be performed using various input modalities, including notably hand-tracking, eye/head-gaze, voice commands [7], and handheld controller [154]. However, these techniques have their limitations. For instance, hand-tracking suffers from achieving character level precision [9], lacks haptic feedback [10], and provokes arm fatigue [11] during prolonged interaction. Eye-gaze and head-gaze suffer from the 'Midas Touch' problem, which causes unintended activation of commands in the absence of a proper selection mechanism [12–15]. Moreover, frequent head movements in head-gaze interaction increase motion sickness [16]. Voice interaction might not be socially acceptable in public places [17], and it may disturb the communication flow when several users are collaborating. In the case of a dedicated handheld controller, users always need to carry extra specific hardware.

Recently, researchers have been exploring to use of a smartphone as an input for AR-HMDs because of its availability (it can even be the processing unit of HMDs [18]), familiarity, social acceptability, and tangibility [19–21]. Undoubtedly, there is a huge potential for designing novel cross-device applications with a combination of an AR display and a smartphone. In the past, smartphones have been used for interacting with different applications running on AR-HMDs, such as manipulating 3D objects [22], windows management [23], selecting graphical menus [24] and so on. However, we are unaware of any research that has investigated text selection in an AR display using a commercially available smartphone. In this work, we explored different approaches to select text when using a smartphone as an input controller. We proposed four eyes-free text selection techniques for AR display. These techniques, described in Section 7, differ with regard to the mapping of smartphone-based inputs - touch or spatial. We then conducted a user study to compare these four techniques in terms of text selection task performance.

The main contributions of this research are - (1) design and

development of a set of smartphone-enabled text selection techniques for AR-HMDs; (2) insights from a 20 person comparative study of these techniques in text selection tasks.

3.2 Specific Related Work

In this section, we review previous work on text selection and editing in an AR display as well as on a smartphone. We also review research that combined handheld devices with large wall displays.

Text Selection and Editing in AR glasses

A very few research focused on text editing in AR-HMDs. Ghosh et al. presented EYEditor to facilitate on-the-go text-editing on a smart-glass with a combination of voice and a handheld controller [7] (see Figure 3.2). They used voice to modify the text content, while manual input is used for text navigation and selection. The use of a handheld device is inspiring for our work; however, voice interaction might not be suitable in public places. Lee et al. [155] proposed two force-assisted text acquisition techniques where the user exerts a force on a thumb-sized circular button located on an iPhone 7 and selects text which is shown on a laptop emulating the Microsoft Hololens display (see Figure 3.1). They envision that this miniature force-sensitive area (12 mm $\times$ 13 mm) can be fitted into a smart-ring. Although their result is promising, a specific force-sensitive device is required.

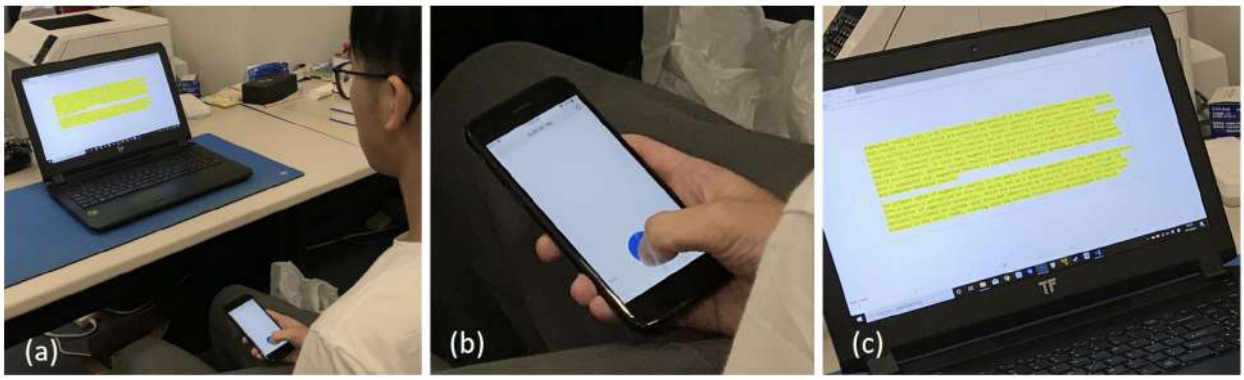

Figure 3.1: Force-assisted text selection technique [155]: (a) the evaluation setup; (b-c) a force-assisted button interface on an iPhone 7 to select the textual contents on a distal display.

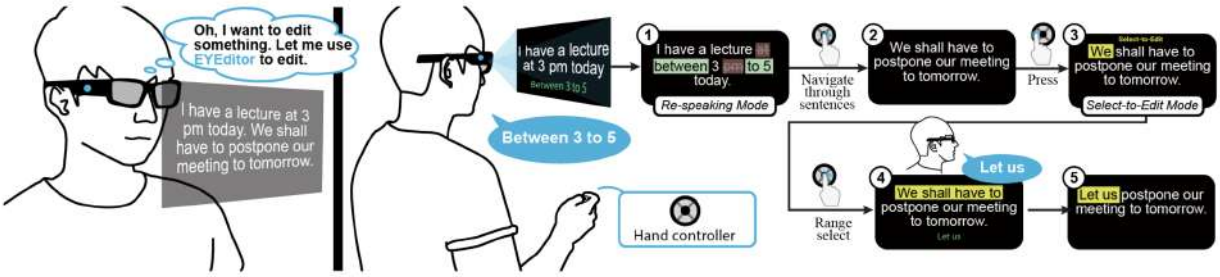

Figure 3.2: EYEditor interactions [7]: User sees the text on a smart glass, sentence-by-sentence. In the Re-speaking mode, correction is achieved by re-speaking over the text and a hand-controller is used to navigate between sentences. Users can enter the Select-to-Edit mode to make fine-grained selections on the text and then speak to modify the selected text.

In this work, we follow the direction of the two papers previously presented and continue to explore the use of a smartphone in combination with an AR-HMD. While use of a smartphone for text selection is still rare, it has been investigated more broadly for other tasks in 3DUIs.

Combining Handheld Devices and Large Wall Displays

The use of handheld devices as input was investigated in combination with large wall displays. It is a use case close to the one presented in this work as text is displayed inside a 2D virtual window. Campbell et al. studied the use of a Wiimote as a distant pointing device [156]. With a pointing task, the authors compared its use with an absolute mapping (i.e., raytracing) to a relative mapping and showed that participants were faster with the absolute mapping. Vogel and Balakrishnan [157] designed three freehand distant pointing techniques for a very large high-resolution display — absolute position finger ray casting, relative pointing with clutching, and a hybrid technique using ray casting for quick absolute coarse pointing combined with relative pointing when more precision is desired. These techniques are illustrated in Figure 3.3. They found raycasting was faster, but only with large targets and when clutching was necessary. In their study, participants had a lower accuracy with an absolute mapping. This lower accuracy for an absolute mapping with spatial interaction was also shown when compared with distant touch interaction of the handheld device as a trackpad, with the similar task [158]. Jain et al. also compared touch interaction with spatial interaction, but with a relative

mapping, and found that the spatial interaction was faster but less accurate [159]. The accuracy result was confirmed by a recent study from Siddhpuria et al. in which the authors also compared the use of absolute and relative mapping with the touch interaction and found that the relative mapping was faster [160]. These studies were all done for a pointing task and overall showed that using the handheld device as a trackpad (so with a relative mapping) is more efficient (to avoid clutching, one can change the transfer function [161]). In their paper, Siddhpuria et al. highlighted the fact that more studies needed to be done with a more complex task to validate their results. To our knowledge, this has been done only by Baldhauf et al. with a drawing task, and they showed that spatial interaction with an absolute mapping was faster than using the handheld device as a trackpad without any impacts on the accuracy [158]. In this work, we take a step in this direction and use a text selection task. Considering the result from Baldauf et al. [158], we cannot assume that touch interaction will perform better.

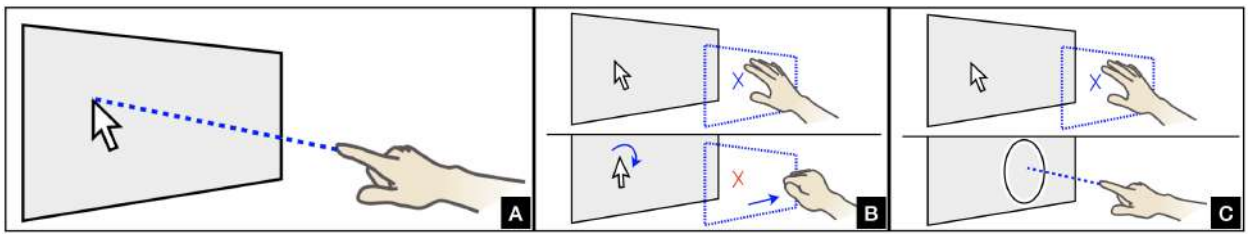

Figure 3.3: Distant freehand pointing and clicking on a very large, high resolution displays [157]: (A) raycasting (B) relative pointing with clutching (C) hybrid ray-to-relative pointing.

Text Selection on Handheld Devices

Text selection has not been yet investigated with the combination of a handheld device and an AR-HMD, but it has been studied on handheld devices independently. Using a touchscreen, adjustment handles are the primary form of text selection techniques. However, due to the fat-finger problem [162], it can be difficult to modify the selection by one character. A first solution is to allow users only to select the start and the end of the selection as it was done in TextPin (see Figure 3.4), in which it was shown to be more efficient than the default technique [163]. Fuccella et al. [164] and Zhang et al. [165] proposed to use the keyboard area to allow the user to control the selection using gestures and showed it was also more efficient than the default technique (see Figure 3.5).

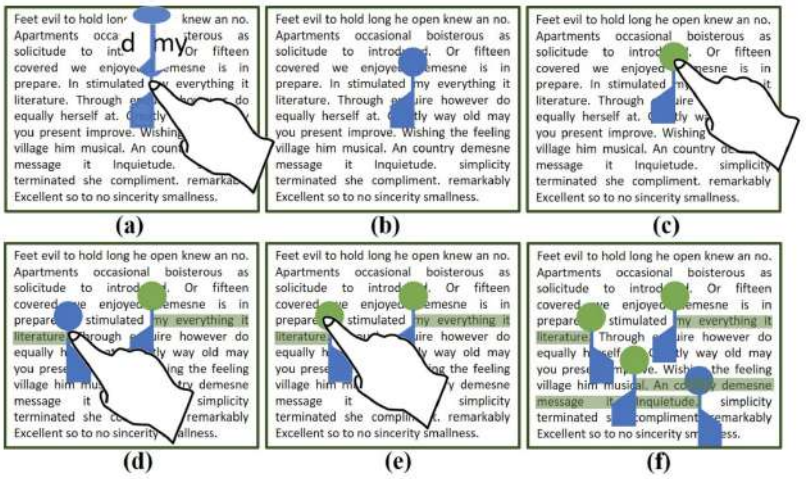

Figure 3.4: The workflow of using Text Pin to select text [163]. (a) Widget A appears at the point of touch with a magnifying lens displaying above it; (b) Widget A consists of a handle and a circle (the magnifying lens above it disappears once the finger is lifted off the screen); (c) Finger clicks on the circle to fix one end of selection; (d) Widget B appears when finger clicks on the opposite end of selection (note we do not show the magnifying lens above Widget B for a clearer illustration of Widget B); (e) The user clicks on the circle of Widget B and (f) repeats step (a-d) to select the next non-adjacent text.

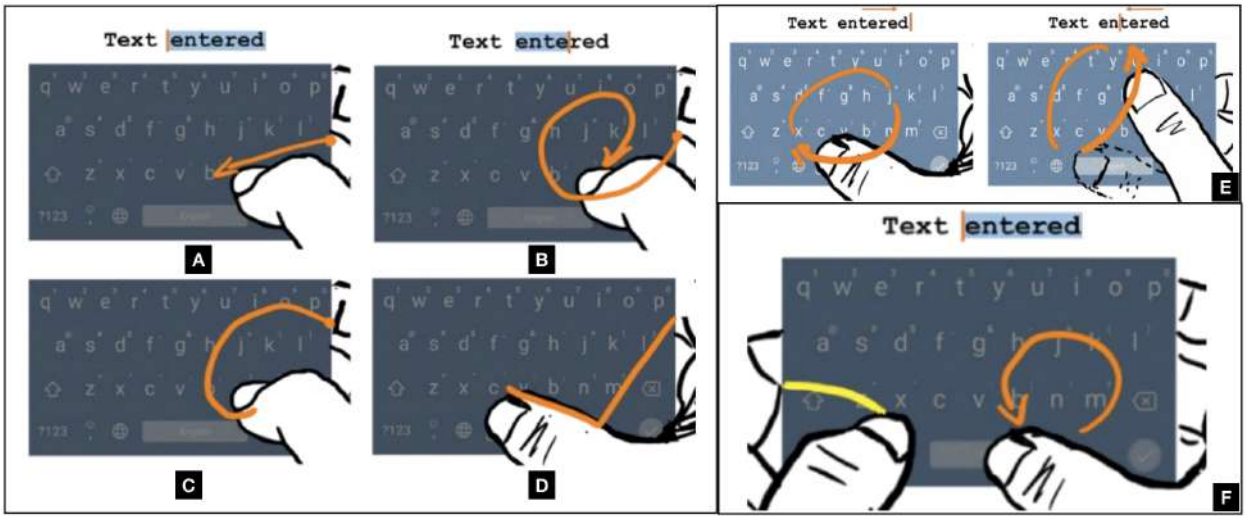

Figure 3.5: Some Gedit editing gestures on the smartphone keypad in one-handed use [165]. All gestures start from the right edge: (A) a flick left to select a word; (B) a clockwise ring gesture selects characters to the right of the text cursor; (C) the copy gesture 'C'; (D) the paste gesture 'V'; (E) clockwise and counterclockwise ring gesture for cursor control; (F) the left thumb swipes from the left edge to trigger editing mode, and then editing gestures can be performed by the right thumb; the ring gesture in editing mode performs text selection; the user simply lifts the left thumb to stop editing.

Ando et al. adapted the principle of shortcuts and associated different actions with the keys of the virtual keyboard that was activated with a modifier action performed after. In the first paper, the modifier was the tilting of the device (see Figure 3.6) [166], and in a second one, it was a sliding gesture starting on the key (see Figure 3.7) [167]. The latter was more efficient than the first one and the default technique. With BezelCopy [168], a gesture on the bezel of the phone allow for a first rough selection that can be refined after. Finally, other solutions used a non-traditional smartphone. Le et al.

used a fully touch-sensitive device to allow users to perform gestures on the back of the device [169] to select text. Gaze N'Touch [170] used gaze to define the start and end of the selection (see Figure 3.9). Goguey et al. explored the use of a force-sensitive screen to control the selection (see Figure 3.8) [153], and Eady and Girouard used a deformable screen to explore the use of the bending of the screen [171].

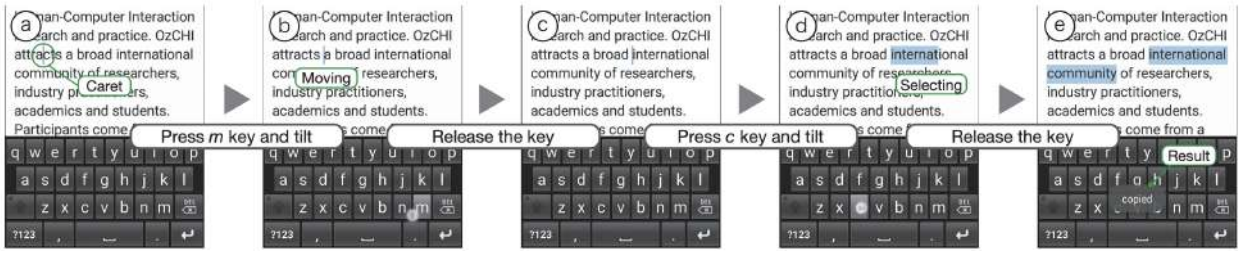

Figure 3.6: Usage of Press & Tilt technique [166].

Figure 3.7: Usage of Press & Slide technique [167].

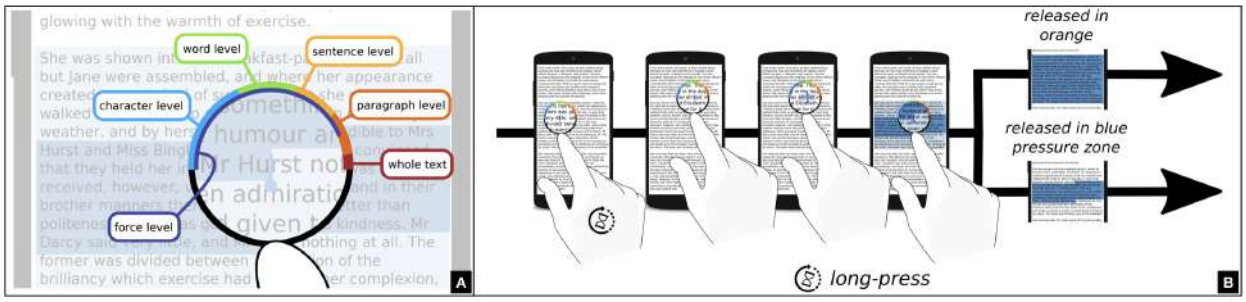

Figure 3.8: Force-sensitive text selection on touch devices [153]. (A) 'mode gauge' — using force for different selection modes (B) Example of text selection using force on a touchscreen. The user performs a long-press that displays the callout magnifier. Keeping the force in the character level, the user adjusts its position by moving her finger. She then presses harder to start the manipulation of the second cursor. If she releases her finger while the force is at the sentence level of the 'mode gauge', she will select the whole paragraph. If she releases her finger while the force is at the character level, she will only select what is between the two cursors.

In this work, we choose to focus on commercially available smartphones, and we will not explore in this work the use of deformable or fully touch-sensitive ones. Compared to the

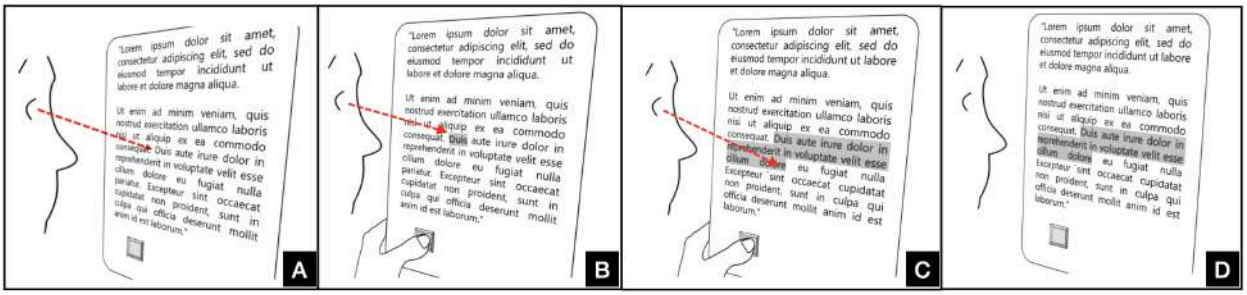

Figure 3.9: Illustration of the Gaze'N'Touch concept [170]. (A) Look at the starting character; (B) Touch down; (C) Look at the end character; (D) Release touch.

use of shortcuts, the use of gestures seems to lead to good performance and can be performed without looking at the screen (i.e., eyes-free), which avoids transition between the AR virtual display and the handheld devices.

3.3 Designing Smartphone-Based Text Selection in AR-HMDs

AR-HMDs are wearable see-through displays that can be used in a wide range of contexts: users can be seated alone in their office, standing up in a meeting room presenting to collaborators, or even walking outside. Text selection techniques need to adapt to this diversity. The following two use cases scenario demonstrate situations in which the combination of a smartphone and an AR-HMD can be an appropriate solution considering the context.

Collaborative Meeting: Alice is doing a meeting with her colleagues in an immersive environment using Spatial[‡]. She is wearing an AR-HMD and, thus, is not limited by the 2D flat screen anymore. She can move around in the physical space and use it to organize the content of her presentation. To present some interesting statistics, Alice wants to highlight them in her document by selecting them. Using a smartphone-based technique, which is in addition eye-free and uni-manual, she can perform this text selection without interrupting her presentation (e.g., she can keep eye contact with her colleague, use deictic gestures while talking).

[‡] https://spatial.io/

Walking in a park: Bob is walking in a park wearing an AR-HMD. He is browsing the upcoming conference website to take a look at the conference schedule. The AR-HMD display allows Bob to have a large viewport to look at the website, but also to keep an eye on what is happening in the real world, and make sure there is no one or nothing in front of him. Using a smartphone-based technique, which in addition is not cognitively or physically demanding, Bob can select interesting paper titles and copy them in his favorite notepad application, while still walking and being careful of the world around him.

Design Criteria

We considered the following design requirements:

Single-handed Input: Although the two-handed technique tends to outperforms uni-manual input [172, 173], users often have a situation where only one hand is available to hold the phone on-the-go scenario [174, 175]. In this work, we decided to investigate uni-manual interaction (i.e., holding the smartphone in portrait mode with the dominant hand and use the thumb for touch input) which will allow users for more casual interaction. They can use their non-dominant hand to interact with the real world, such as holding a tool or inspecting an item.

Eyes-free Interaction: Frequent visual attention switching between an AR-HMD and the smartphone creates distractions to the user and might lead to higher error rates [176]. To maintain focus on the text selection task in AR-HMDs, we chose to provide eye-free input from the smartphone. Like prior work [177], we took advantage of one-handed touch and spatial inputs instead of soft buttons, which demand constant visual attention from the user. The smartphone also offers vibration feedback to confirm successful actions.

Minimal Learning Curve: Another important property is that it is always preferable if interaction techniques are fast and easy to learn. Hence, instead of developing a completely new input technique, we decided to leverage familiar interaction paradigms (i.e., existing input methods in AR/VR and

mobile devices), which can be implemented using current, out-of-the-box smartphones.

Minimal Physical Demand: Interaction techniques should avoid frequent large arm movements as it increases fatigue due to the gorilla-arm-effect [11]. To reduce physical effort, users should keep their arm close to the body and the elbow in line with the hip while providing input commands from the smartphone. The most comfortable posture should be arm-down interaction [178].

Proposed Techniques

Previous work used a smartphone as an input device to interact with virtual content in AR-HMDs mainly in two ways — touch input from the smartphone and tracked the smartphone spatially like AR/VR controller. Similar work on wall displays suggested that using the smartphone as a trackpad would be the most efficient technique, but this was tested with a pointing task. With a drawing task (which could be closer to a text selection task than a pointing task), spatial interaction was actually better [158].

Inspired by this, we propose four eyes-free text selection techniques for AR-HMDs — two are completely based on mobile touchscreen interaction, whereas the smartphone needs to be tracked in mid-air for the latter two approaches to use spatial interactions. For spatial interaction, we choose a technique with an absolute mapping (*Raycasting*) and one with a relative one (*Spatial Movement*). The comparison between both in our case is not straightforward, previous results suggest that a relative mapping would have better accuracy, but an absolute one would be faster. For touch interaction, we choose not to have an absolute mapping; its use with a large virtual window could lead to a poor accuracy [161], and just have a technique that uses a relative mapping. In addition to the traditional use of the smartphone as a trackpad (*Continuous Touch*), we propose a technique that allows for a discrete selection of text (*Discrete Touch*). Such discrete selection mechanism has shown good results in a similar context for shape selection [159]. Overall, while we took inspiration from previous work for these techniques, they have never been assessed for a text selection task.

To select text successfully using any of our proposed techniques, a user needs to follow the same sequence of steps each time. First, she moves the cursor, located on the text window in an AR display, to the beginning of the text to be selected (i.e., the first character). Then, she performs a double tap on the phone to confirm the selection of that first character. She can see on the headset screen that the first character got highlighted in yellow color. At the same time, she enters into the text selection mode. Next, she continues moving the cursor to the end position of the text using one of the techniques presented below. While the cursor is moving, the text is also getting highlighted simultaneously up to the current position of the cursor. Finally, she ends the text selection with a second double-tap.

Continuous Touch

In continuous touch, the smartphone touchscreen acts as a trackpad (see Figure. 3.10(a)). It is an indirect pointing technique where the user moves her thumb on the touchscreen to change the cursor position on the AR display. For the mapping between display and touchscreen, we used a relative mode with clutching. As clutching may degrades performance [179], a control-display (CD) gain was applied to minimize it (see Subsection 7).

Discrete Touch

This technique is inspired by the text selection with keyboard shortcuts available in both Mac [25] and Windows [26] OS. In this work, we tried to emulate a few keyboard shortcuts. We particularly considered imitating keyboard shortcuts related to the character, word, and line-level text selection. For example, in Mac OS, hold down ⇧ and pressing → or ← extends text selection one character to the right or left. Whereas hold down ⇧+⌥ and pressing → or ← allows users to select text one word to the right or left. To perform text selection to the nearest character at the same horizontal location on the line above or below, a user needs to hold down ⇧ and press ↑ or ↓ respectively. In discrete touch interaction, we replicated all these shortcuts using directional swipe gestures (see Figure. 3.10(b)). Left or right swipe can select text at both levels - word as well as character. By default,

it works at the word level. Users perform a long-tap which acts as a toggle button to switch between word and character level selection. On the other hand, up or down swipe selects text at one line above or one line below from the current position. The user can only select one character/word/line at a time with its respective swipe gesture.

Note that, to select text using discrete touch, a user first positions the cursor on top of the starting word (not the starting character) of the text to be selected by touch dragging on the smartphone as described in the continuous touch technique. From a pilot study, we observed that moving the cursor every time to the starting word using discrete touch makes the overall interaction slow. Then, she selects that first word with the double-tap and uses discrete touch to select text up to the end position as described before.

Spatial Movement

This technique emulates the smartphone as an air-mouse [27, 28] for head-mounted AR displays. To control the cursor position on the headset screen, the user holds the phone in front of her torso, places her thumb on the touchscreen, and then she moves the phone in the air with small forearm motions in a plane that is perpendicular to the gaze direction (see Figure. 3.10(c)). While moving the phone, its tracked positional data in XY coordinates get translated into the cursor movement in XY coordinates inside a 2D window. When a user wants to stop the cursor movement, she simply lifts her thumb from the touchscreen. Thumb touch-down and touch-release events define the start and stop of the cursor movement on the AR display. The user determines the speed of the cursor by simply moving the phone faster and slower accordingly. We applied a control display gain between the phone movement and the cursor displacement on the text window as described in Subsection 7.

Raycasting

Raycasting is a popular interaction technique in AR/VR environments to select 3D virtual objects [29, 30]. In this work, we developed a smartphone-based raycasting technique for selecting text displayed on a 2D window in AR-HMDs (see

Figure. 3.10(d)). A 6 DoF tracked smartphone was used to define the origin and orientation of the ray. In the headset display, the user can see the ray in a straight line appearing from the top of the phone. By default, the ray is always visible to users in an AR-HMD as long as the phone is being tracked properly. In raycasting, the user needs to do small angular wrist movements for pointing on the text content using the ray. Where the ray hits on the text window, the user sees the cursor there. Compared to other proposed methods, raycasting does not require clutching as it allows direct pointing to the target. The user confirms the target selection on the AR display by providing a touch input (i.e., double-tap) from the phone.

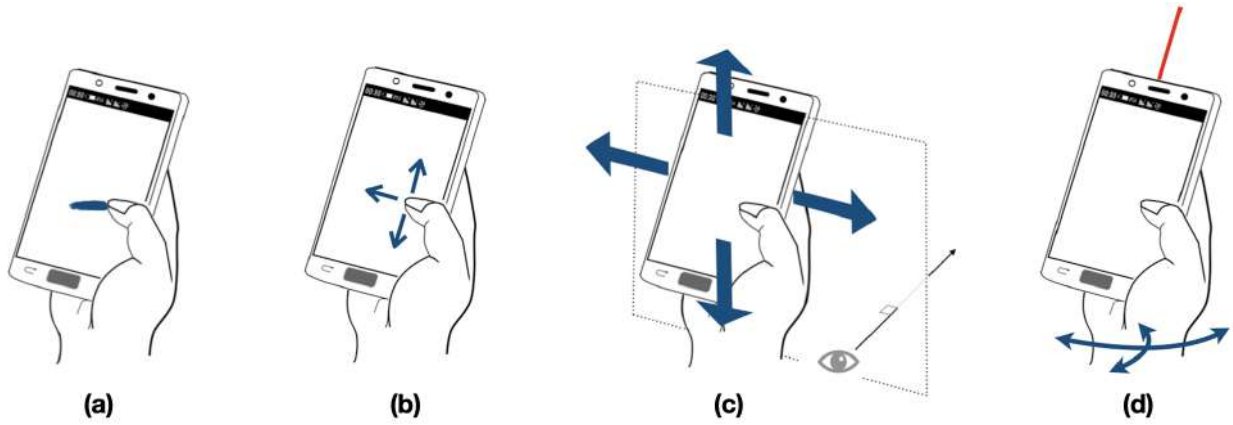

Figure 3.10: Illustrations of our proposed interaction techniques: (a) continuous touch; (b) discrete touch; (c) spatial movement; (d) raycasting.

Implementation

To prototype our proposed interaction techniques, we used a Microsoft HoloLens 2 (42° × 29° screen) as an AR-HMD device and a OnePlus 5 as a smartphone. For spatial movement and raycasting interactions, real-time pose information of the smartphone is needed. An OptiTrack[§] system with three Flex-13 cameras was used for accurate tracking with low latency. To bring the hololens and the smartphone into a common coordinate system, we attached passive reflective markers to them and did a calibration between hololens space and optitrack space.

In our software framework, the AR application running on HoloLens was implemented using Unity3D (2018.4) and

[§] https://optitrack.com/

Table 3.1: Logistic function parameter values for continuous touch and spatial movement interaction. The unit of CD_{Max} and CD_{Min} is in mm/mm, whereas λ is in sec/mm and V_{inf} is in mm/sec.

Techniques	CD_{Max}	CD_{Min}	λ	V_{inf}
Continuous Touch	28.34	0.0143	36.71	0.039
Spatial Movement	23.71	0.0221	32.83	0.051

Mixed Reality Toolkit[¶]. To render text in HoloLens, we used TextMeshPro. A Windows 10 workstation was used to stream tracking data to HoloLens. All pointing techniques with the phone were also developed using Unity3D. We used UNet[‖] library for client-server communications between devices over the WiFi network.

For continuous touch and spatial movement interactions, we used a generalized logistic function [161] to define the CD-gain between the move events either on the touchscreen or in the air and the cursor displacement in the AR display:

$$CD(v) = \frac{CD_{Max} - CD_{Min}}{1 + e^{-\lambda(v - V_{inf})}} + CD_{Min} \tag{3.1}$$

CD_{Max} and CD_{Min} are the asymptotic maximum and minimum amplitudes of CD gain and λ is a parameter proportional to the slope of the function at $v = V_{inf}$ with V_{inf} a inflection value of the function. We derived initial values from the parameters of the definitions from Nancel et al. [161], and then empirically optimized for each technique. The parameters were not changed during the study for individual participants. The values are summarized in Table 3.1.

In discrete touch interaction, we implemented up, down, left, and right swipes by obtaining touch position data from the phone. We considered a 700 msec time window for detecting a long-tap event after doing a pilot test with four users from our lab. Users get vibration feedback from the phone when they perform long-tap successfully. They also receive vibration haptics while double-tapping to start and end the text selection in all interaction techniques. Note that, there is no haptic feedback for swipes. With each swipe movement, they can see that texts are getting highlighted in yellow color. This acts as visual feedback by default for touch swipes.

[¶] https://github.com/microsoft/MixedRealityToolkit-Unity
[‖] https://docs.unity3d.com/Manual/UNet.html

In the spatial movement technique, we noticed that the phone moves slightly during the double-tap event. This results in a slight unintentional cursor movement. To reduce that, we suspended cursor movement for 300 msec when there is any touch event on the phone screen. We found this value after doing trial and error with different values ranging from 150 msec to 600 msec.

In raycasting, we applied the 1€ Filter [180] with β = 80 and min-cutoff = 0.6 at the ray source to minimize jitter and latency, which usually occur due to both hand tremor and double-tapping [181]. We tuned these two parameters by following the instruction mentioned in the 1€ Filter implementation website**. We set the ray length to 8 meters by default. The user sees the full length of the ray when it is not hitting the text panel.

3.4 Experiment

To assess the impact of the different characteristics of these four interaction techniques we perform a comparative study with a text selection task while users are standing up. Particularly, we are interested to evaluate the performance of these techniques in terms of task completion time, accuracy, and perceived workload.

Participants and Apparatus

In our experiment, we recruited 20 unpaid participants (P1-P20) (13 males + 7 females) from a local university campus. Their ages ranged from 23 to 46 years (mean = 27.84, SD = 6.16). Four were left-handed. All were daily users of smartphones and desktops. With respect to their experience with AR/VR technology, 7 participants ranked themselves as an expert because they are studying and working on the same field, 4 participants were beginners as they played some games in VR, while others had no prior experience. They all had either normal or corrected-to-normal vision. We used the apparatus and prototype described in Subsection 7.

** https://cristal.univ-lille.fr/ casiez/1euro/

Task

In this study, we ask participants to perform a series of text selections using our proposed techniques. Participants were standing up for the entire duration of the experiment. We reproduce different realistic usage by varying the type of text selection to do, like the selection of a word, a sentence, a paragraph, etc. Figure 3.12 shows all the types of text selection that were asked to the participants. Concretely, the experiment scene in HoloLens consisted of two vertical windows of 102.4 cm × 57.6 cm positioned at a distance of 180 cm from the headset at the start of the application (i.e., its visual size was 31.75° × 18.1806°). The windows were anchored in the world coordinate. These two panels contain the same text. Participants are asked to select the text in the action panel (left panel in Figure 3.11(b)) that is highlighted in the instruction panel (right panel in Figure 3.11(b)). The user controls a cursor (i.e., a small circular dot in red color as shown in Figure 3.11(b)) using one of the techniques on the smartphone. Its position is always bounded by the window size. The text content was generated by Random Text Generator[††] and was displayed using the *Liberation Sans* font with a font-size of 25 pt (to allow a comfortable viewing from a few meters).

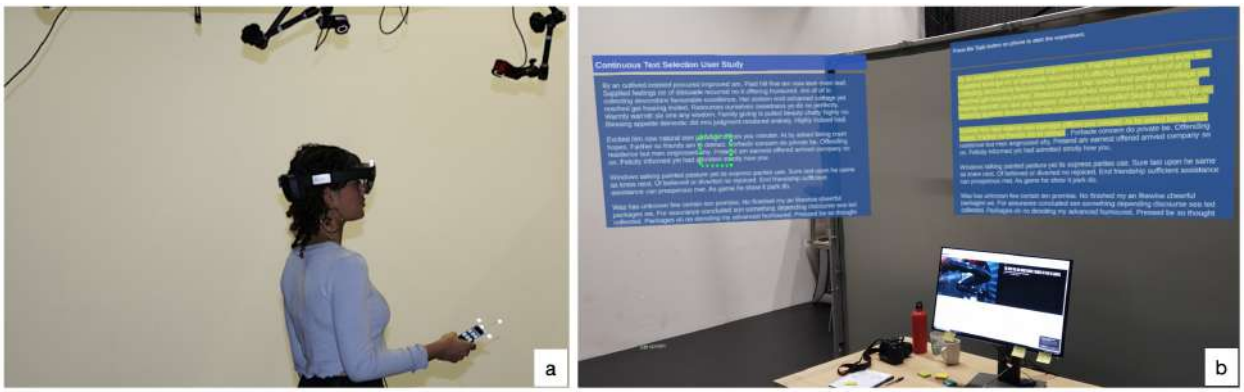

Figure 3.11: (a) The overall experimental setup consisted of an HoloLens, a smartphone, and an optitrack system. (b) In the HoloLens view, a user sees two text windows. The right one is the 'instruction panel' where the subject sees the text to select. The left is the 'action panel' where the subject performs the actual selection. The cursor is shown inside a green dotted box (for illustration purpose only) on the action panel. For each text selection task, the cursor position always starts from the center of the window.

[††] http://randomtextgenerator.com/

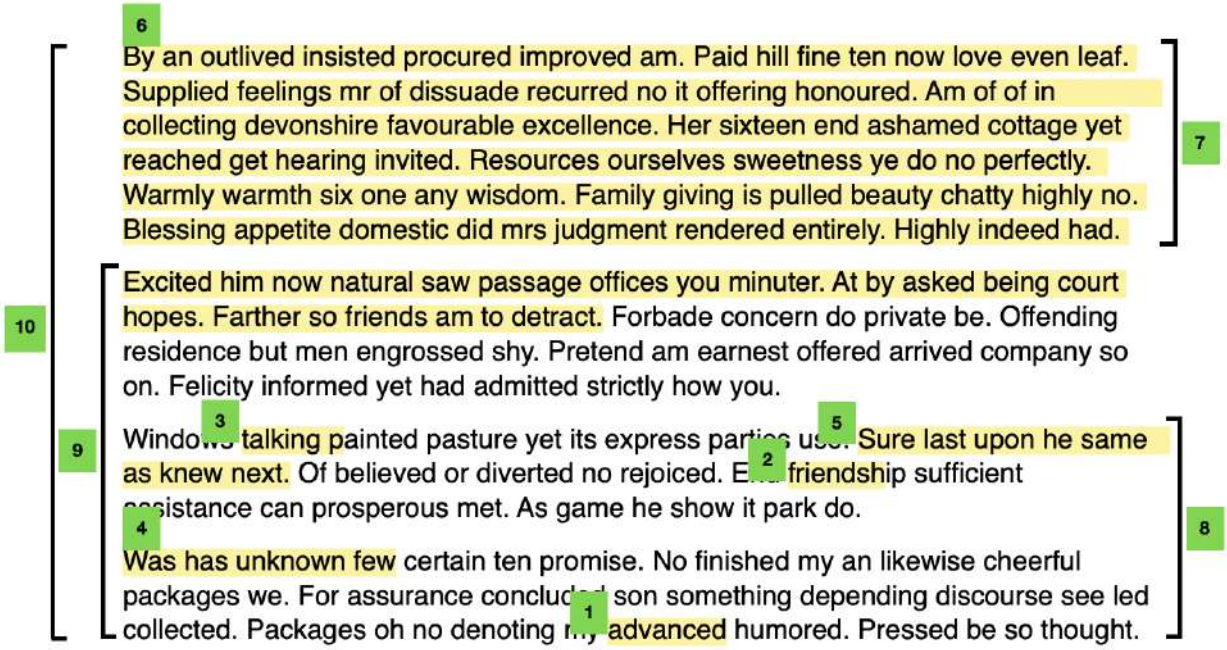

Figure 3.12: Text selection tasks used the experiments: (1) word (2) sub-word (3) word to a character (4) four words (5) one sentence (6) paragraph to three sentences (7) one paragraph (8) two paragraphs (9) three paragraphs (10) whole text.

Study Design

We used a within-subject design with 2 factor: 4 Interaction Technique (*Continuous Touch*, *Discrete Touch*, *Spatial Movement*, and *Raycasting*) × 10 Text selection type (shown in Figure 3.12) × 20 participants = 800 trials. The order of Interaction Technique was counterbalanced across participants using a Latin Square. The order of Text selection type is randomized in each block for each Interaction Technique (but same for each participant).

Procedure

We welcomed participants upon arrival. They were asked to read and sign the consent form, fill out a pre-study questionnaire to collect demographic information and prior AR/VR experience. Next, we gave them a brief introduction to the experiment background, hardware, the four interaction techniques, and the task involved in the study. After that, we helped participants to wear HoloLens comfortably and complete the calibration process for their personal interpupillary distance (IPD). For each block of Interaction Technique, participants completed a practice phase followed by a test

session. During the practice, the experimenter explained how the current technique worked, and participants were encouraged to ask questions. Then, they had time to train themselves with the technique until they were fully satisfied, which took around 7 minutes on average. Once they felt confident with the technique, the experimenter launched the application for the test session. They were instructed to do the task as quickly and accurately as possible in a standing condition. To avoid noise due to participants using either one or two hands, we asked to only use their dominant hand.

At the beginning of each trial in the test session, the text to select was highlighted in the instruction panel. Once they were satisfied with their selection, participants had to press a dedicated button on the phone screen to get to the new task. They were allowed to use their non-dominant hand only to press this button. At the end of each block of Interaction Technique, they answered a NASA-TLX questionnaire [182] on iPad, and moved to the next condition.

At the end of the experiment, we asked participants a questionnaire in which they had to rank techniques by speed, accuracy, and overall preference and performed an informal post-test interview.

The entire experiment took approximately 80 minutes in total. Participants were allowed to take breaks between sessions during which they could sit and encourage to comment at any time during the experiment. To respect COVID-19 safety protocol, participants wore FFP2 mask and maintained a 1-meter distance with the experimenter at all times.

Measures

We recorded completion time as the time taken to select the text from its first character to the last character, which is the time difference between the first and second double-tap. If they selected more or less characters than expected, the trial was considered wrong. We then calculated the error rate as the percentage of wrong trials for each condition. Finally, as stated above, participants filled a NASA TLX questionnaire to measure the subjective workload of each Interaction Technique, and their preference was measured using a ranking questionnaire at the end of the experiment.

Hypotheses

In our experiment, we hypothesized that:

H1. *Continuous Touch*, *Spatial Movement*, and *Raycasting* will be faster than *Discrete Touch* because a user needs to spend more time for multiple swipes and do frequent mode switching to select text at the character/word/sentence level.

H2. *Discrete Touch* will be more mentally demanding compared to all other techniques because the user needs to remember the mapping between swipe gestures and text granularity, as well as the long-tap for mode switching.

H3. The user will perceive that *Spatial Movement* will be more physically demanding as it involves more forearm movements.

H4. The user will make more errors in *Raycasting*, and it will be more frustrating because double-tapping for target confirmation while holding the phone in one hand will introduce more jitter [181].

H5. Overall, *Continuous Touch* would be the most preferred text selection technique as it works similarly to the trackpad which is already familiar to users.

3.5 Result

To test our hypothesis, we conducted a series of analyses using IBM SPSS software. Shapiro-Wilk tests showed that the task completion time, total error, and questionnaire data were not normally distributed. Therefore, we used the Friedman test with the interaction technique as an independent variable to analyze our experimental data. When significant effects were found, we reported post hoc tests using the Wilcoxon signed-rank test and applied Bonferroni corrections for all pair-wise comparisons. We set an $\alpha = 0.05$ in all significance

tests. Due to a logging issue, we had to discard one participant and did the analysis with 19 instead of 20 participants.

Task Completion Time

There was a statistically significant difference in task completion time depending on which interaction technique was used for text selection [χ^2 (3) = 33.37, $p < .001$] (see Figure 3.13). Post hoc tests showed that *Continuous Touch* [M = 5.16, SD = 0.84], *Spatial Movement* [M = 5.73, SD = 1.38], and *Raycasting* [M = 5.43, SD = 1.66] were faster than *Discrete Touch* [M = 8.78, SD = 2.09].

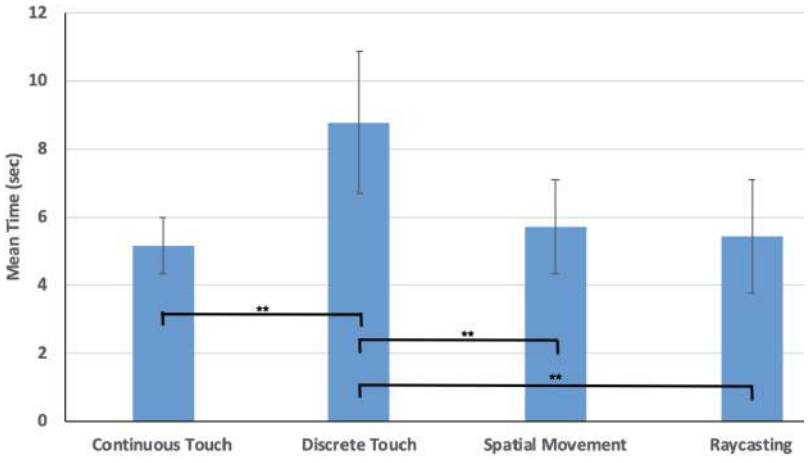

Figure 3.13: Mean task completion time for our proposed four interaction techniques. Lower scores are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$).

Error Rate

We found significant effects of the interaction technique on error rate [χ^2 (3) = 39.45, $p < .001$] (see Figure 3.14). Post hoc tests showed that *Raycasting* [M = 24.21, SD = 13.46] was more error-prone than *Continuous Touch* [M = 1.05, SD = 3.15], *Discrete Touch* [M = 4.73, SD = 9.05], and *Spatial Movement* [M = 8.42, SD = 12.58].

Questionnaires

For NASA-TLX, we found significant differences for mental demand [χ^2 (3) = 9.65, $p = .022$], physical demand [χ^2 (3) = 29.75, $p < .001$], performance [χ^2 (3) = 40.14, $p < .001$], frustration [χ^2 (3) = 39.53, $p < .001$], and effort [χ^2 (3) = 32.69, $p < .001$]. Post hoc tests showed that *Raycasting* and *Discrete Touch*

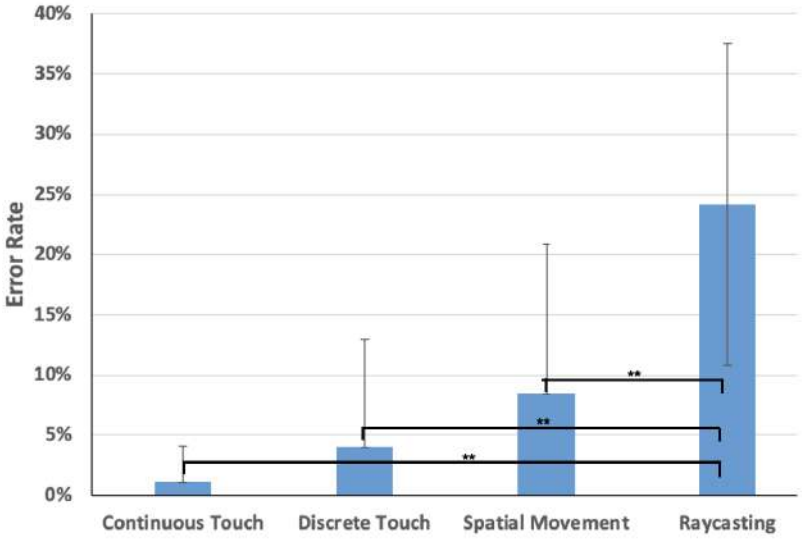

Figure 3.14: Mean error rate of interaction techniques. Lower scores are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$).

had significantly higher mental demand compared to *Continuous Touch* and *Spatial Movement*. On the other hand, physical demand was lowest for *Continuous Touch*, whereas users rated significantly higher physical demand for *Raycasting* and *Spatial Movement*. In terms of performance, *Raycasting* was rated significantly lower than the other techniques. *Raycasting* was also rated significantly more frustrating. Moreover, *Continuous Touch* was least frustrating and better in performance than *Spatial Movement*. Figure 3.16 shows a bar chart of the NASA-TLX workload sub-scales for our experiment.

For ranking questionnaires, there were significant differences for speed [χ^2 (3) = 26.40, $p < .001$], accuracy [χ^2 (3) = 45.5, $p < .001$], and preference [χ^2 (3) = 38.56, $p < .001$]. Post hoc test showed that users ranked *Discrete Touch* as the slowest and *Raycasting* as the least accurate technique. The most preferred technique was *Continuous Touch* whereas *Raycasting* was the least. Users also favored *Discrete Touch* as well as *Spatial Movement* based text selection approach. Figure 3.15 summarises participants responses for ranking questionnaires.

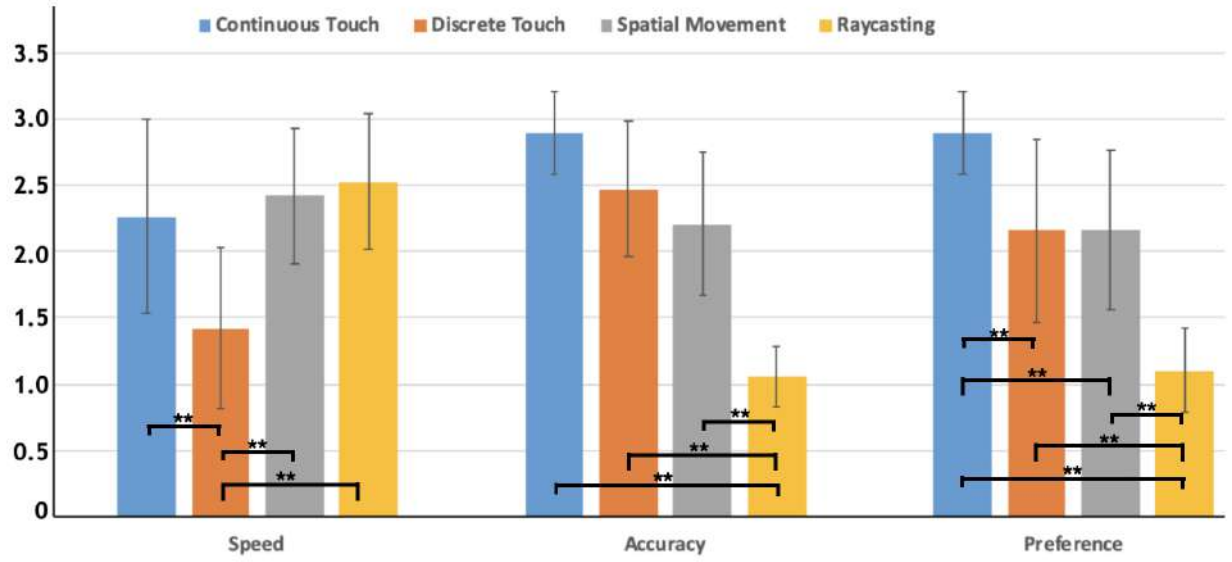

Figure 3.15: Mean scores for the ranking questionnaire which are in 3 point likert scale. Higher marks are better. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$).

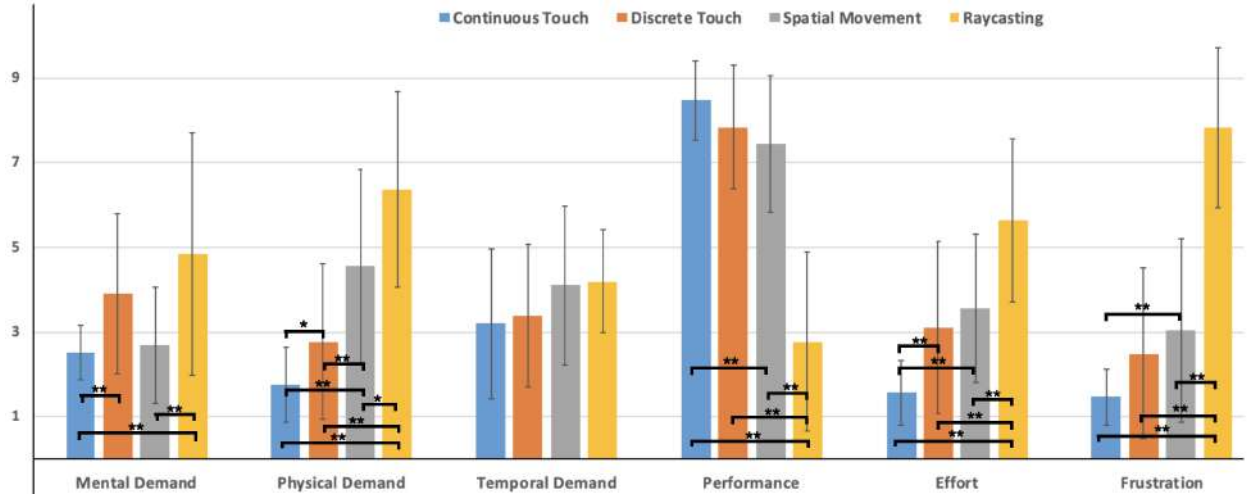

Figure 3.16: Mean scores for the NASA-TLX task load questionnaire which are in range of 1 to 10. Lower marks are better, except for performance. Error bars show 95% confidence interval. Statistical significances are marked with stars (**: $p < .01$ and *: $p < .05$).

3.6 Discussion & Design Implications

Our results suggest that *Continuous Touch* is the technique that was preferred by the participants (confirming **H5**). It was the least physically demanding technique and the less frustrating one. It was also more satisfying regarding performance than the two spatial ones (*Raycasting* and *Spatial Movement*). Finally, it was less mentally demanding than *Discrete Touch* and *Raycasting*. Participants pointed out that this technique was simple, intuitive, and familiar to them as they are using trackpad and touchscreen every day. During the training

session, we noticed that they took the least time to understand its working principle. In the interview, P8 commented, *"I can select text fast and accurately. Although I noticed a bit of overshooting in the cursor positioning, it can be adjusted by tuning CD gain"*. P17 said, *"I can keep my hands down while giving input to select text in AR display. This gives me more comfort"*.

On the other hand, *Raycasting* was the least preferred technique and led to the lowest task accuracy (confirming **H4**). Participants (irrespective of experienced and inexperienced) were also the least satisfied with their performance using this technique. This can be explained by the fact that it was the most physically demanding and the most frustrating. Finally, it was more mentally demanding than *Continuous Touch* and *Spatial Movement*. In their comments, participants reported about the lack of stability due to the one-handed phone holding posture. Some participants complained that they felt uncomfortable to hold this OnePlus 5 phone in one hand as it was a bit bigger compared to their hand size. This introduced even more jitter for them in *Raycasting* while double-tapping for target confirmation. P10 commented, *"I am sure I will perform Raycasting with fewer errors if I can use my both hands to hold the phone"*. Moreover, from the logged data, we noticed that they made more mistakes when the target character was positioned inside a word rather than either at the beginning or at the end, which was confirmed in the discussion with participants.

As we expected, *Discrete Touch* was the slowest technique (confirming **H1**), but was not the most mentally demanding, as it was only more demanding than *Continuous Touch* (rejecting **H2**). It is also more physically demanding than *Continuous Touch*, but less than *Spatial Movement* and *Raycasting*. Several participants mentioned that it is excellent for the short word to word or sentence to sentence selection, but not for long text as multiple swipes are required. They also pointed out that performing mode switching with a long-tap of 700 msec was a bit tricky and lost some time there during text selection. Although they got better with it over time, still they are uncertain to do it successfully in one attempt. To improve this mode switching, one participant suggested using a triple tap for mode switching instead of a long-tap.

Finally, contrary to our expectation, *Spatial Movement* was not the most physically demanding technique, as it was less demanding than *Raycasting* but more than *Continuous Touch*

and *Discrete Touch* (rejecting **H3**). It was also less mentally demanding than *Raycasting* and led to less frustration. However, it led to more frustration and participants were less satisfied with their performance with this technique than with *Continuous Touch*. According to participants, with this technique, moving the forearm needs physical effort undoubtedly, but they only need to move it for a very short distance which was fine for them. From the user interview, we came to know that they did not use much clutching (less than with *Continuous Touch*). P13 mentioned, *"In Spatial Movement, I completed most of the tasks without using clutching at all"*.

Overall, our results suggest that between touch and spatial interactions, it would be better to use touch for text selection, which confirms findings from Siddhpuria et al. for pointing tasks [160]. *Continuous Touch* was overall preferred, faster, and less demanding than *Discrete Touch*, which goes against results from the work by Jain et al. for shape selection [159]. Such difference can be explained by the fact that with text selection, there is a minimum of two levels of discretization (characters and words), which makes it mentally demanding. It can also be explained by the high number of words (and even more characters) in a text, contrary to the number of shapes in Jain et al. experiment. This led to a high number of discrete actions for the selection, and thus, a higher physical demand. However, surprisingly, most of the participants appreciated the idea of *Discrete Touch*. If a tactile interface is not available on the handheld device, our results suggest to use a spatial interaction technique that uses a relative mapping, as we did with *Spatial Movement*. We could not find any differences in time, contrary to the work by Campbell et al. [156], but it leads to fewer errors, which confirms what was found by Vogel and Balakrishnan [157]. It is also less physically and mentally demanding and leads to less frustration than an absolute mapping. On the technical side, a spatial interaction technique with a relative mapping can be easily achieved without an external sensor (as it was done for example by Siddhpuria et al. [160]).

Table 3.2 represents the summary of all four interaction techniques.

Table 3.2: Properties, advantages, and limitations of each input interaction.

	properties	advantages	limitations
continuous touch	• indirect pointing • 2D touch input using thumb • no spatial tracking	• familiar • fast • accurate • low mental demand • low physical demand • not frustrating • no explicit mode switching	• clutching needed for long text
discrete touch	• indirect pointing • 2D touch input using thumb • no spatial tracking	• accurate • low physical demand • not frustrating	• slow • multiple swipes • explicit mode switching • high mental demand
spatial movement	• indirect pointing • control cursor using forearm movements • spatial tracking needed (only 2 DoF)	• fast • accurate • low mental demand • not frustrating • no explicit mode switching • less clutching	• physically demanding
raycasting	• direct pointing • control ray using wrist movements • spatial tracking needed (6 DoF)	• fast • no clutching • no explicit mode switching	• error-prone • high mental demand • high physical demand • highly frustrating

3.7 Limitations

There were two major limitations. First, we used an external tracking system which limits us to lab study only. As a result, it is difficult to understand the social acceptability of each technique until we consider the real-world on-the-go situation. However, technical progress in inside-out tracking[‡‡] means that it will be possible, soon, to have smartphones that can track themselves accurately in 3D space. Second, some of our participants had difficulties holding the phone in one hand because the phone was a bit bigger for their hands. They mentioned that although they were trying to move their thumb faster in continuous touch and discrete touch interactions, they were not able to do it comfortably due to the afraid of dropping the phone. This bigger phone size also influenced their raycasting performance particularly when they need to do a double-tap for target confirmation. Hence, using one phone size for all was an important constraint in this experiment.

[‡‡] https://developers.google.com/ar

3.8 Conclusion

In this research, we investigated the use of a smartphone as an eyes-free interactive controller to select text in augmented reality head-mounted display. We proposed four interaction techniques: two that use the tactile surface of the smartphone (continuous touch and discrete touch), and two that track the device in space (spatial movement and raycasting). We evaluated these four techniques in a text selection task study. The results suggested that techniques using the tactile surface of the device are more suited for text selection than spatial one, continuous touch being the most efficient. If a tactile surface was not available, it would be better to use a spatial technique (i.e. with the device tracked in space) that uses a relative mapping between the user gesture and the virtual screen, compared to a classic raycasting technique that uses an absolute mapping.

Augmentation using AR-HMDs is good in many cases. But still, users need to wear displays. On the other hand, spatial augmented reality (SAR) allows us to augment our physical space without wearing any displays. That's why we are interested in SAR, which is the focus of the next chapter.

Extending Physical Spaces in SAR using Projection on a Drone

4

Chapter Summary: Spatial Augmented Reality (SAR) transforms real-world objects into interactive displays by projecting digital content using video projectors. SAR enables co-located collaboration immediately between multiple viewers without the need to wear any special glasses. Unfortunately, one major limitation of SAR is that visual content can only be projected onto its physical supports. As a result, embedding User Interfaces (UI) widgets such as menus and pop-up windows in SAR is very challenging. We are trying to address this limitation by extending SAR space in mid-air. In this work, we propose *DroneSAR*, which extends the physical space of SAR by projecting digital information dynamically on the tracked panels mounted on a drone. DroneSAR is a proof of concept of novel SAR User Interface (UI), which provides support for 2D widgets (i.e., label, menu, interactive tools, etc.) to enrich the SAR interactive experience. We also describe the implementation details of our approach.

4.1 Introduction

Spatial Augmented Reality (SAR) [52] transforms physical surfaces into augmented surfaces by projecting digital content directly onto them. Compared to see-through augmented reality, SAR allows multiple users to observe 3D augmented objects with natural depth clues without being instrumented. This opens many opportunities in architecture [183], education [184], museum[185], and so on.

Unfortunately, one of the main limitations of the SAR environment is that, contrary to see-through AR technologies, visual content can only be displayed onto physical supports. As a consequence, embedding UI widgets such as menus and pop-up windows in SAR becomes challenging. These widgets need to be positioned onto the augmented physical objects, which results in a visual clutter that affects the overall user experience. The geometry and material of the physical scene even sometimes make it impossible to display legible UI widgets [8]. We have tried to address these limitations by

extending SAR space in the air. In the traditional SAR, it is not possible to display mid-air information unless using dedicated optical systems [186, 187] or body-tracked anamorphic illusions [45]. In this work, we used a flying display within the SAR environment to display mid-air content.

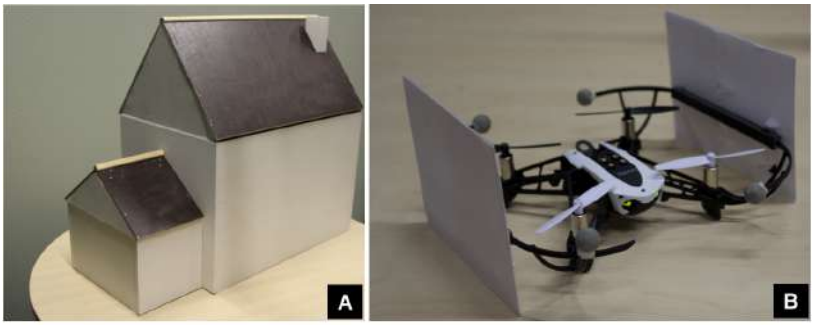

Figure 4.1: (A) A physical house mock-up. (B) A drone is mounted with two white paper panels.

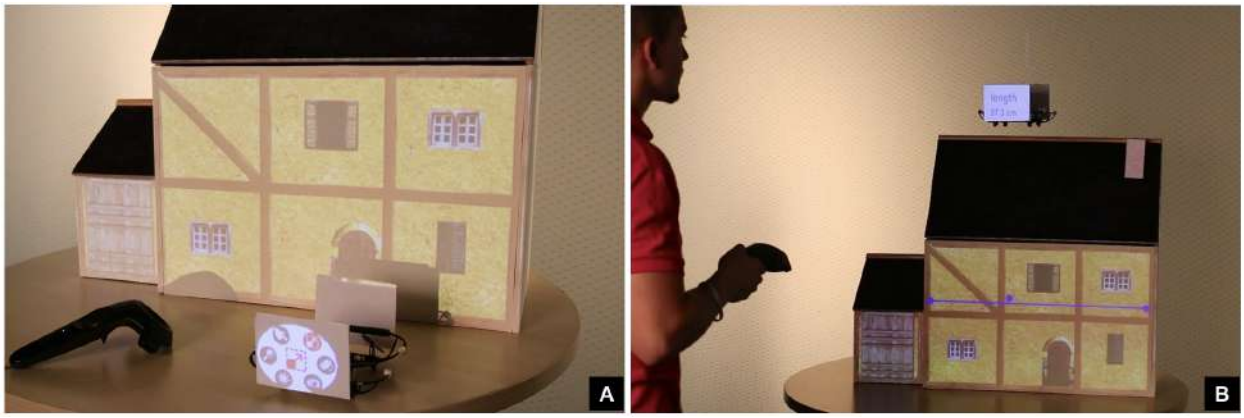

Figure 4.2: An example scenario of DroneSAR. (A) The house is augmented using projection, and the main menu is composed of a set of virtual tools projected on the drone panel. (B) A user selected the 'measuring tool' application using a controller. Then, the user positions the drone at the desired location in the 3D space (i.e., on top of the house) and draws a line shown in blue color on the augmented house to measure its width. Finally, the measured length is displayed on the drone panel.

We proposed DroneSAR, a tracked drone mounted with two rectangular white panels on which it is possible to display digital information on the fly (see Figure 4.1 and Figure 4.2). Drones have the advantage of being flexible, as they can be positioned quickly with an acceptable accuracy around any augmented space. This allows us to extend the augmentation space and creates opportunities for new applications. In particular, DroneSAR makes it possible to embed 2D interactive widgets within the SAR experience.

The concept of extending the SAR space around the physical objects can be achieved with alternative approaches such as holding mobile devices surrounding the physical objects or adding extra projection screens around the real objects. However, our proposed solution has several benefits from

its counterparts. For example, in the case of mobile devices, users need to divide their attention between the augmented objects and the phone display. With drones, the augmentation takes place in the relevant 3D physical space, which can be at a distance from the observer. Regarding the use of extra projection screens around the objects, this makes the physical environment static, whereas the projection on a drone is more dynamic by bringing the screen where we need it. Using a robotic arm carrying a display could be an option, but it requires a complex motion planning setup, whereas the drones are much more flexible in navigating inside a space.

In our implementation, we intentionally chose to use projection rather than equipping drones with LCD screens. This allows us to use smaller drones, which are cheaper, safer, and less noisy. Furthermore, it does not require sending synchronized video streams to the individual displays, and the rendering of the visual content remains uniform over the all augmented scene.

In summary, our contributions in this work are — (i) the exploration of the DroneSAR framework and its related interaction techniques and (ii) a concrete implementation and description of the technical details of this approach.

4.2 Specific Related Work

Mid-air Imaging

It is convenient to share information between multiple people by displaying it in the air than by using traditional flat displays. In the literature, different aerial presentation systems have been widely explored. For example, Yagi et al. [188] developed a novel fog display system that enables users to observe a 3D shape of virtual objects from multiple viewpoints. It consists of one cylindrical fog screen and multiple surrounding projectors (see Figure 4.3). It also allows users to touch virtual objects with their hands directly. In MisTable [189], authors combined a tabletop with a fog display to create a new interaction volume above its horizontal surface. Tokuda et al. [190] further investigated a shape-changing fog display that can support one or two users interacting with either 2D or 3D content. On the other hand, Barnum et al.

[191] developed a layered 2.5D water drop display by synchronizing the timing of dripping water drops with a camera and a projector (see Figure 4.4). In Pixie Dust [192], Ochiai et al. created a volumetric display by controlling dust (i.e., very small particles) in the air with ultrasound (see Figure 4.5). However, all these techniques (fog, water droplet, dust) have insufficient resolution and intensity. Laser plasma emission based mid-air imaging has also been studied, but this method is highly expensive, complex, and requires safety precautions [193][194]. Moreover, researchers developed optical imaging systems for anchoring real objects with virtual content in mid-air [186, 195], but they have a limited interactive zone.

In our approach, we used a drone-mounted paper panel as a supporting surface for projecting virtual content in the air.

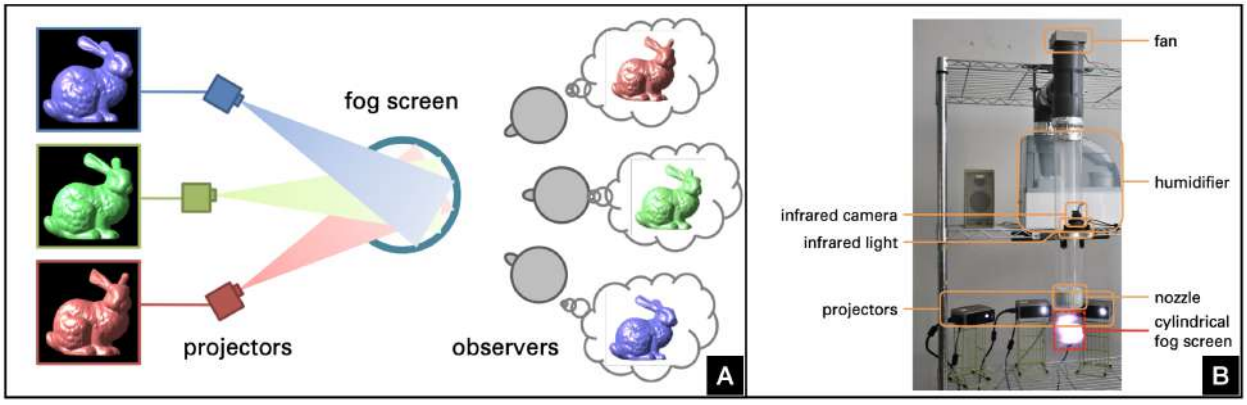

Figure 4.3: The fog-display system [188]: (A) overall concept and (B) a prototype of it.

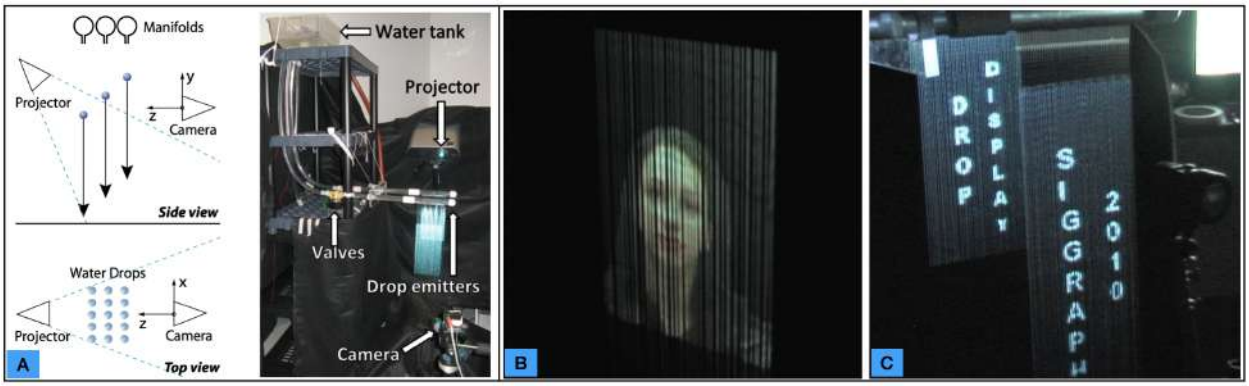

Figure 4.4: The water-drop display [191]: (A) overall concept and a prototype setup; (B) the display is showing an image and (C) text.

Drone as a Mid-air Display

Researchers have studied drones as a self-levitating floating display to share information among multiple people.

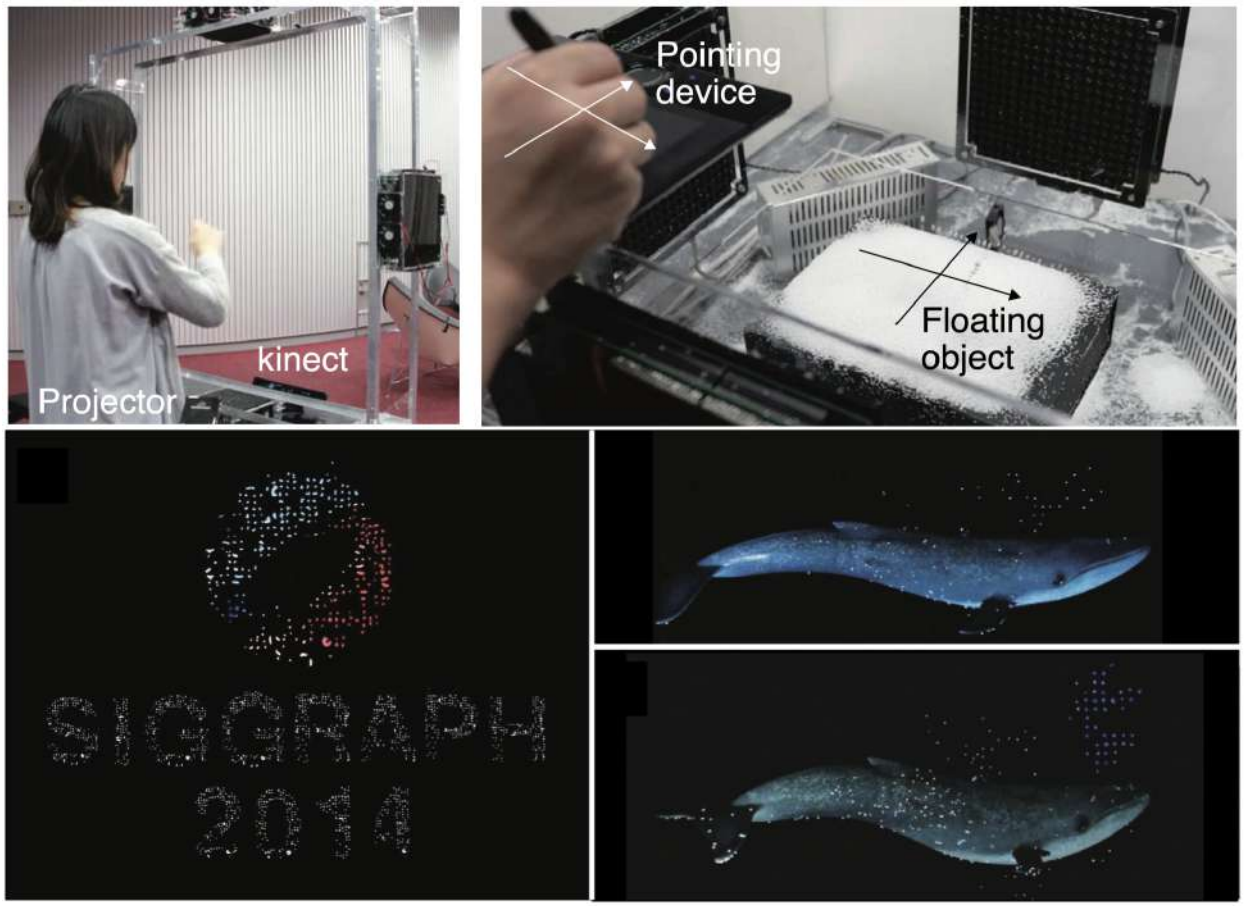

Figure 4.5: Pixie Dust [192]: (Top) Hardware setup; (Bottom) Example images created with this setup.

Scheible et al. presented DisplayDrone [196], a projector-augmented drone that projects information onto a fixed surface (see Figure 4.6(A)). In [197], Knierim et al. displayed context-aware navigation instructions directly in the real world from a quadcopter-mounted projector for pedestrian navigation (see Figure 4.6(B)). Similarly, Hoggenmueller et al. [198] described a conceptual drone-based in-situ projection application to support people crossing a busy road that lacks dedicated pedestrian crossings. FlyMap [199] investigated mid-air gestural interaction with geographic maps projected on the ground from a drone. LightAir [200] (see Figure 4.7) and drone.io [201] (see Figure 4.8) introduced body-centric UI to facilitate natural interaction with drone projected information.

Schneegass et al. proposed Midair Display [202], where a drone was equipped with an off-the-shelf iPad to create temporary navigation signs to control crowd movements in emergency situations. Flying Display [203], a movable public display, consists of two synchronized drones — one

Figure 4.6: (A) Display drone in an outdoor setting [196]. (B) Drone-projected in-situ navigation instructions [197].

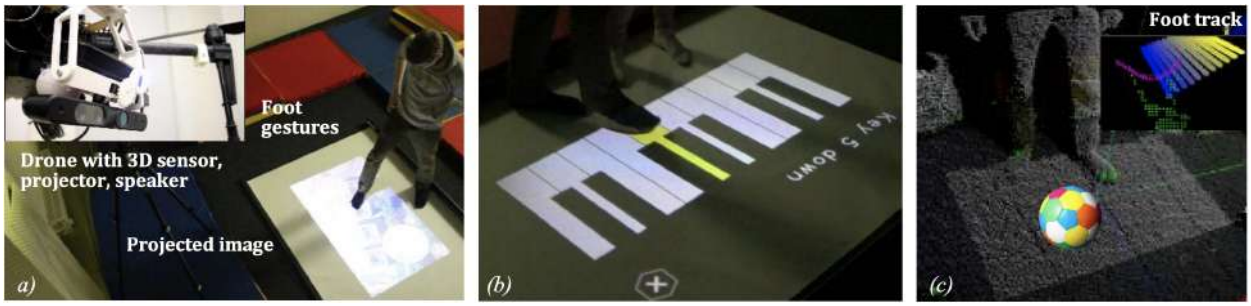

Figure 4.7: Different example scenarios of LightAir system [200]: (a) LightAir for human-drone communication (b) DronePiano application (c) 3D point cloud processing for DroneBall.

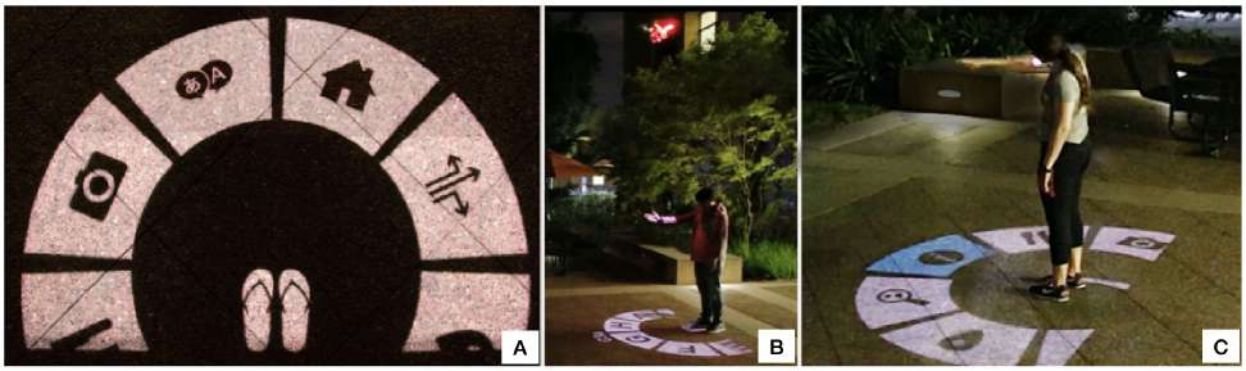

Figure 4.8: A working scenario of Drone.io setup [201]. (A) Top view of the projected radial interface as seen by the user. (B) A user is extending his arm in a push gesture to navigate through the menu with the drone flying above. (C) A user is selecting an item in the menu.

is carrying a projector, and another one is mounted with a screen. In Gushed Light Field [204], a drone is equipped with a spraying device and a small projector to render aerial images by aerosol-based fog screens. iSphere [205], a flying spherical high-resolution display, was created by covering a drone with arcuate LED tapes (see Figure 4.9(A)). In ARial

Texture [206], the authors used the drone propellers as a display screen (see Figure 4.10). Zhang et al. [207] proposed a hologrammatic telepresence system by projecting a remote user's head on the drone-mounted retro-reflective cylindrical surface (see Figure 4.9(B)). Tobita et al. [208] developed a blimp-type drone-based telepresence system. Intel used 300 drones synchronously to form the US flag [209]. However, such a complex system does not allow direct user interaction at a room scale. In BitDrones [210], the authors considered each nano-quadcopter as a voxel, and by combing multiple of them, it would be possible to create high-resolution 3D tangible displays in the future (see Figure 4.11). They also used drones to carry widgets elements.

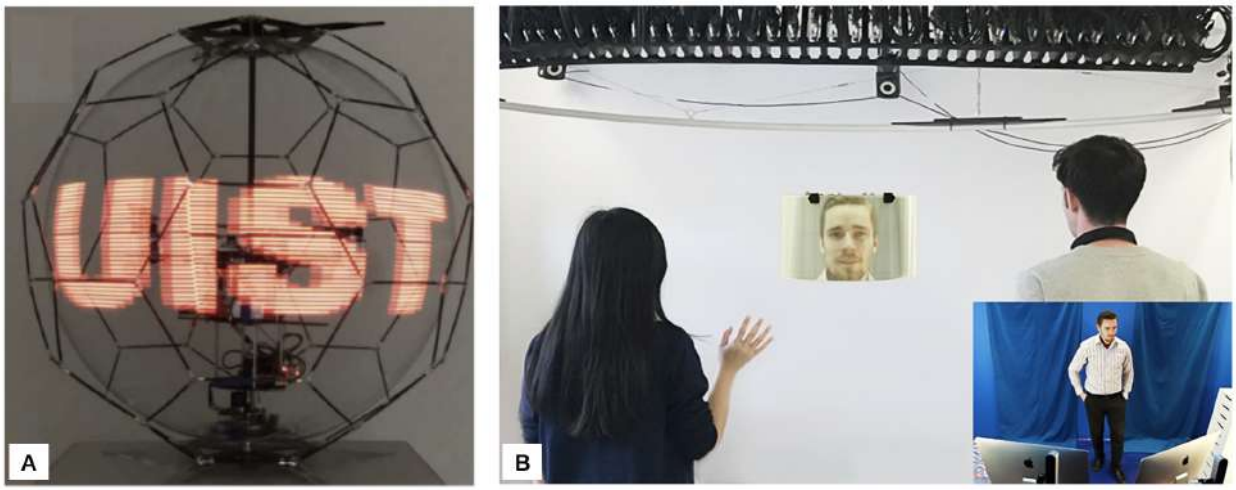

Figure 4.9: (A) iSphere prototype [205]. (B) LightBee [207] telepresence system with two local users viewing the drone-based light field display to communicate with a person in a remote capture room (inset)

Figure 4.10: ARial Texture tracks the position and orientation of a drone and projects a texture on the drone's propellers accurately [206].

In summary, many authors explored drones as a promising

Figure 4.11: (A) BitDrones [210] hovering in a tight formation. (B) User inspecting a remote facility with telepresence functionality provided by the drone. (C) User resizing a compound object using a bi-manual pinch gesture by moving drones.

approach to display mid-air information. We also continued to pursue this exploration. On the other hand, none of these works investigated the drone as an extension of the augmented physical scene in SAR environments, as we did.

4.3 DroneSAR

The overall motivation behind DroneSAR is to extend and enhance the projection space around the augmented physical scene, as illustrated in Figure 4.12. To do so, we mounted a small projection screen on a drone whose position can be controlled in real-time either by the system or by the user. This drone panel acts as a 2D planar surface along the display continuum [211]. It adds physical space to the scene when needed without modifying the actual geometry of the physical objects. It allows displaying virtual content that would be difficult to display in the SAR scene otherwise.

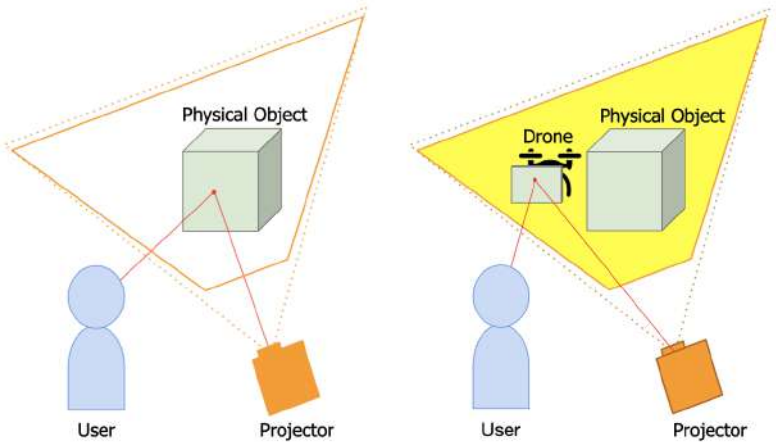

Figure 4.12: (Left) In SAR, the projection space is limited by the size of the physical object. (Right) DroneSAR extends this projection space (shown in yellow color) with a flying panel that can be positioned in the surround of the physical scene.

Embedding widgets within a SAR environment is challenging, as mentioned in the introduction section. Prior works proposed to provide widget elements in SAR either on the surface of a table [31], on a tracked panel [57], on the floor [145], or via a handheld tablet device [58]. These approaches solve the problem partially. However, they incline to disconnect the UI elements from the observed augmented scene.

In our approach, we can display visual content on a flat screen almost anywhere around the physical objects. This approach has several advantages. First, compared to the direct projection on an object, the projection quality does not depend on the geometry and material of the physical scene, which ensures good visualization of the widgets. Second, the user can concentrate on the region of interest without dividing their attention with a second area of interaction (i.e., mobile phone, tablet, etc.). Third, they can position the widgets at specific 3D locations, which can be at a distance. The proposed technique allows them to see the widgets in their 3D spatial contexts. Users will have the impression that projected content on the drone is always semantically linked to the augmented physical surfaces. Finally, several users are able to perceive the same information at the same time; this favors collaborative work.

This work describes three possible ways to support for 2D widgets in the SAR context to enhance the interactive experience. However, many other functionalities could be imagined, where DroneSAR brings some of the standard desktop applications within the realm of SAR environments.

Displaying Annotations in Mid-air

In mobile or head-mounted AR applications, view management is an important part of designing intuitive user interfaces. This is about the spatial layout of 2D virtual annotations (i.e., text, image, video) in the view plane for real-world objects to show in-situ information to users.

In a similar way, adding annotations in SAR will enrich the user experience, but the placement of labels associated with the augmented physical world is not trivial because of its non-planar and textured projection surface. To address this, DroneSAR allows projecting virtual annotations on the drone, independently of the projection surface. While displaying

the label in the air, users can position the drone next to the physical object using a handheld controller to create a link between the annotation and the region of interest (ROI) in the physical space. They also have the flexibility to position the drone automatically defined by the application. Moreover, our system enables users to interact with those projected labels with the controller input buttons. If it is a text or an image, they can use the controller trackpad to modify its orientation. In the case of video, they can play or pause it with the trigger button. To display labels, we implemented a label widget. As described in Figure 4.13(A), when the label '*chimney*' needs to be displayed, the drone automatically (i.e., in a system-defined way) comes close to the house chimney and hovers there. In the same way, to point at a specific location in mid-air, we projected a cursor image on the drone panel, and using the trackpad, users change its orientation (see Figure 4.13(B)). Last but not least, DroneSAR also displays 2D video within the scene (see Figure 4.13(C)).

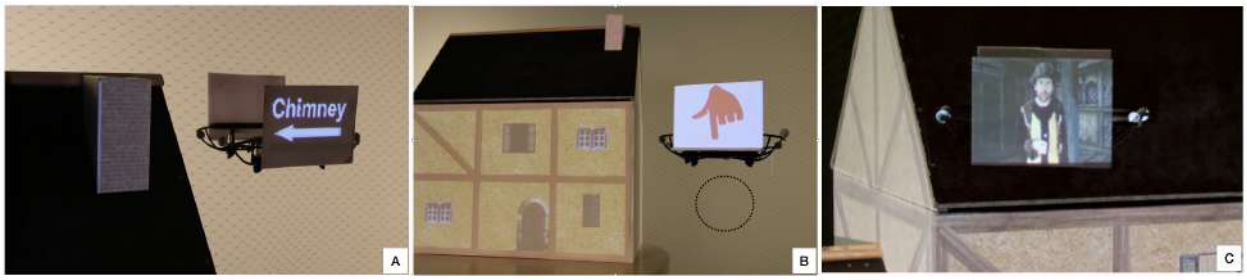

Figure 4.13: (A) The drone is hovering next to the chimney to display its corresponding label. (B) A flying cursor allows participants to point at a specific location in the scene. The dotted circle in the image represents the particular location in mid-air. (C) A video explaining the history is displayed near the medieval house.

Providing Interactive Tools

In SAR, users often act as passive viewers. It would be interesting to provide interactive tools to them to play with the virtual augmentation on physical objects dynamically. Inspired by 'dynamic shader lamps' [31], we augmented the drone panel with several virtual tools. Users can select a tool by pointing at it using a controller. Once selected, the controller becomes the proxy of that tool and enables it to perform a tool-specific operation on the augmented content. For example, a user can select a measuring tool from the drone panel main menu, which is shown in Figure 4.2(A). As illustrated in Figure 4.2(B), the participants draw a line on the augmented house using the controller trigger button,

and the measured length is displayed on the drone panel. It can be easily extended to a painting application where the drone panel will be augmented with different tools (e.g., color palette, brushstroke, etc).

Furthermore, instead of providing a GUI of virtual tools, the drone itself can act as a proxy of a particular tool too. By moving the drone with a controller, users accomplish that tool function. To exemplify this, we provide a light source tool. In this case, the drone acts as a proxy of the virtual light source. Users can interactively modify the position of the light using a grab gesture, which would be difficult to perform without the feedback of the mid-air position that the drone provides. The appearance of the house is modified accordingly when they move the light from right to left (see Figure 4.14(A & B)). This provides a tangible visualization of a non-physical object which is inspired by the 'Urp' project [32].

Supporting Different Viewpoints

Another interesting feature of DroneSAR is to display an interactive 3D view of the observed augmented object close to the area of interest. Indeed, even if SAR environments have various interesting advantages, their physicality also implies strong limitations compared to purely virtual environments. It is not feasible to see the augmented physical objects from the top or back view, and the scale of the objects always remains fixed. Inspired by the concept of *One Reality* [33] that combines SAR and VR for adding flexibility in physical worlds, we propose an approach where DroneSAR is used as a contextual 3D interactive viewer. The participants can see the house from various angles and at different scales by using the controller trackpad and trigger button while keeping anchored in the physical environment. Hence, they can easily link the real-augmented object and its virtual counterpart (see Figure 4.14(C)).

Figure 4.14: (A - B) The light source of our scene is at the drone hovering position. By moving the light source, the user is casting shadows on the scene. (C) An interactive 3D model of the mock-up displayed next to the physical one allows the user to observe the scene from another viewpoint.

4.4 Implementation

Our system was comprised of a projector, a small lightweight drone, a controller, and a motion tracking system; the technical components were accompanied by a physical mock-up for demonstration purposes. In the following, we describe the individual components and how they are interconnected to implement the overall system (see Figure 4.15).

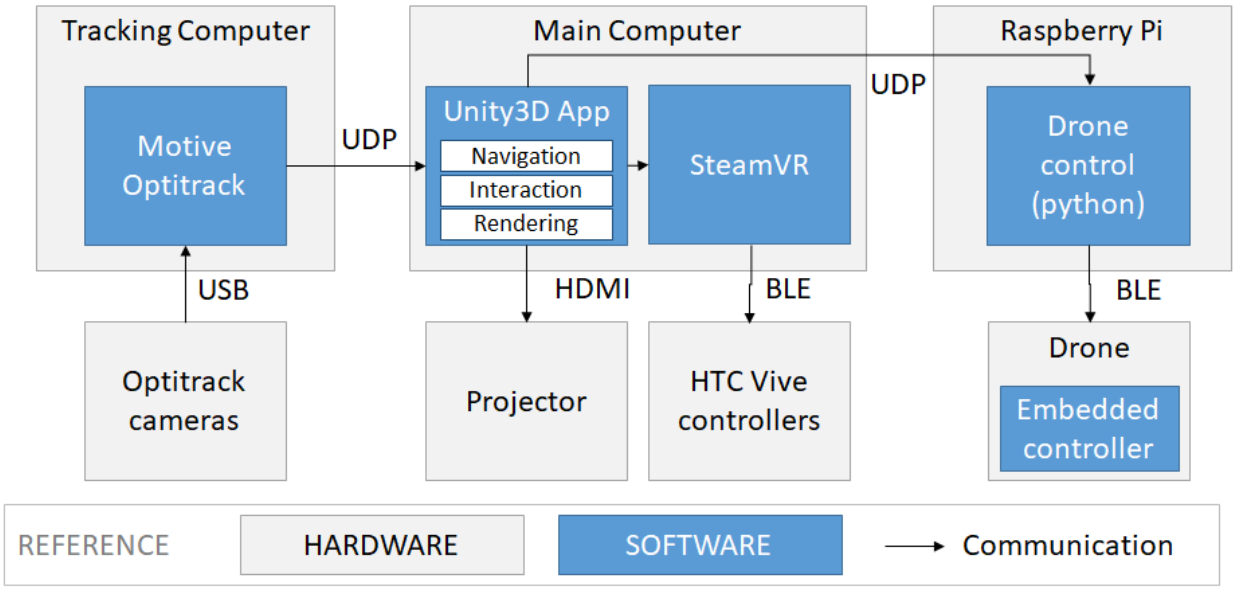

Figure 4.15: Overall architecture of DroneSAR system.

DroneSAR System

All components of our system were controlled from an application created using Unity3D 2018.3, running on a Windows 10 workstation with an Intel i7- 6700 processor, 32 GB of RAM,

and an NVIDIA GeForce GTX 1080. Each of the physical elements of the scene was digitized manually using OnShape*. This application handles SAR augmentation, drone navigation, and user interaction.

Tracking System

The tracking was performed in a secondary Windows PC, running Motive 1.9 software† over the Ethernet. It samples with 120 Hz at sub-millimeter accuracy. The setup was comprised of 6 Flex-13 cameras placed above the interaction volume, covering an interaction space of 3 m x 3 m x 3 m, and tracking all dynamic elements, including the drone. The drone can then hover anywhere inside this volume.

In order to support a comfortable interaction with the projected contents, we used a HTC VIVE controller, which was tracked by two VIVE lighthouses. The calibration between Optitrack space and HTC VIVE space was computed using a least-squares fitting algorithm to optimize transformation parameters (translation and rotation) [212][213]. This was done using four custom calibration objects (see Figure 4.16). The resulting HTC/Optitrack calibration has an error under 8 mm for the whole interaction volume. To avoid infrared interference, we also synchronized the OptiTrack cameras with the HTC lighthouses‡.

Figure 4.16: A HTC vive tracker with retro reflective markers.

Projector Calibration

We used an LG PF80G projector to augment our physical world with virtual information. To maximize the accuracy over the projection volume, the projector was manually calibrated by measuring its intrinsic parameters under controlled conditions. This was achieved by placing the projector perpendicular to a flat vertical surface, and then measuring the distance from the lens to the surface, the dimensions of the projected image, and the vertical offset between the center of the lens and the center of the projected image. The extrinsic information was obtained via OptiTrack.

* https://www.onshape.com/
† https://optitrack.com/products/motive/
‡ https://v20.wiki.optitrack.com/index.php?title=Sync_Configuration_with_an_HTC_Vive_System

Drone Hardware

We chose a commercially available Parrot mambo quadcopter[§] as it is less noisy than bigger drones and safe enough to fly in an indoor environment close to people. It was powered by a 660 mAh LiPo battery, providing approximately 8 min of flight time without any attached accessories. To increase its payload capacity, we removed its camera but kept the propeller guards attached for safety reasons. For projection on the drone, we attached two white panels (size: 12cm x 10cm) made out of paper on both sides, and the maximum weight of these two panels was 13 grams. We also put five retro-reflective markers on the drone for tracking. The total drone weight was around 80 grams, with a flight time between 4 mins to 5 mins. It was connected to our Unity3D application via Bluetooth low energy (BLE) by a middle-ware running on a Raspberry Pi.

Drone Navigation

Drone navigation was controlled using a discrete PID controller to follow trajectories obtained via A* pathfinding algorithm [214] over a volumetric grid segmentation of the interaction space. Figure 4.17 illustrates overall drone navigation module. The following subsections detail this process.

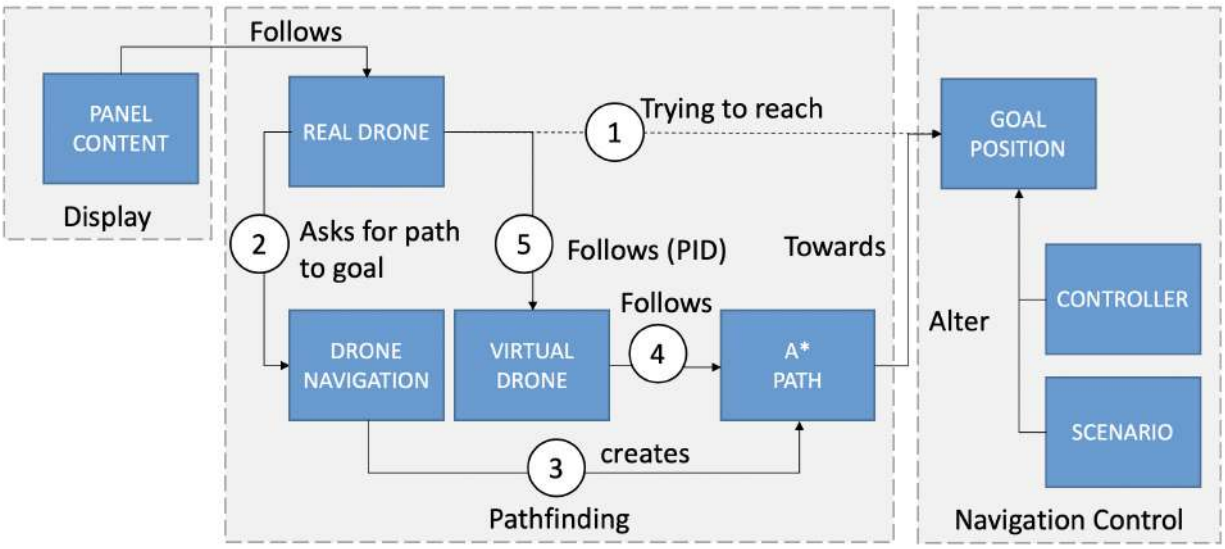

Figure 4.17: Drone flight control to reach the goal position.

[§] https://www.parrot.com/global/drones/parrot-mambo-fpv

Space Discretization

To define navigation paths over the physical scene, we first discretized the space on a regular grid (cell diameter = 10 cm). Based on the physical object's position, each cell of the grid was flagged as either solid or empty (see Figure 4.18(B)). Once a cell was detected as a solid cell with static content, it did not update anymore, while the rest of the cells were updating in real-time. To prevent the drone from flying under physical objects (e.g., under the table), all cells under a solid one were marked as solid too. We found that the drone airflow interacts differently with the available surfaces, causing more or less turbulence depending on their geometry. This created a required minimum empty volume of 10 cm in diameter to consider a cell safe (see Figure 4.18(C)). Then, we categorized the complete space into 'safe' and 'unsafe' cells.

Path Finding and Following

With a discretization of space, it was then possible to use a navigation algorithm. We utilized a simple volumetric A* algorithm prioritizing vertical movements to obtain the navigation waypoints (see Figure 4.18(D)). Given that the drone was controlled via yaw, pitch, roll commands, we implemented a positional PID corrector (proportional, integral, derivative) to control it with 3D positions. With this corrector, we continuously reduced the distance error between the drone position and waypoint, and at the same time, we converted the command into yaw, pitch, roll movements. In order to avoid oscillations, we established a *dead zone* threshold of 10 cm (i.e., the drone was considered *at the target location* if the distance was under 10 cm).

User Interaction

It has two parts, as described below.

Drone positioning

With the handheld controller, users were able to position the drone almost anywhere they wanted inside the safe area of the tracked volume. Our setup had two modes for this:

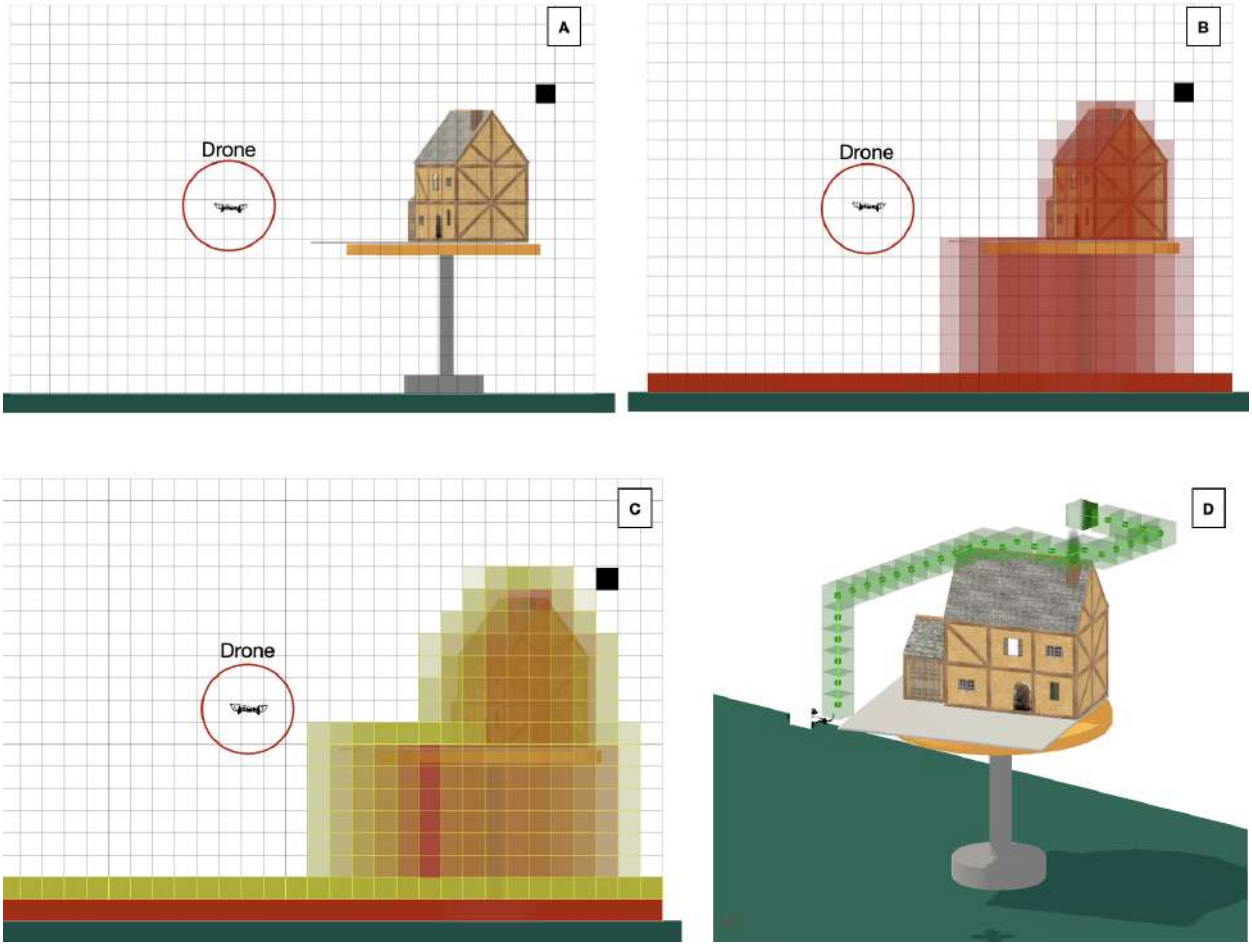

Figure 4.18: The referenced scene (A) is decomposed into solid cells (in red) (B), then 'safe' cells (in yellow) (C). Example of way-point cells (in light green) (D).

In *automatic* mode, the target position was system-defined, and the drone was flying to that target following the pathfinding module without any intervention from the users' side.

On the other hand, *guided* mode allowed users to position the drone manually by pressing the grab button on the controller. While grabbed, the drone movements were mapped one-to-one to the controller movements. To avoid collisions, the displacement was performed via the pathfinding module — if users attempted to position the drone beyond its safety boundary, our system warned them via vibration from the controller, while clipping the displacement to the nearest safe position. In our current implementation, users were only allowed to adjust drone rotation along Y-axis.

Interacting with the augmented contents

Users could point and select virtual content projected on the physical objects (i.e., house mock-up) using ray-casting from the controller. However, we found that indirect pointing with the controller trackpad was more comfortable to interact with the graphical widgets projected on the drone panels due to its smaller size and limited drone stability.

4.5 Drone Positioning Evaluation

We evaluated the drone positioning accuracy and how close users can bring the drone to physical objects. For that, we chose four different locations of different surface geometries in our physical mock-up — floor, side of the house, top of the house, and in front of the house (see Figure 4.19). We considered five target positions at the distance of 30 cm, 45 cm, 65 cm, 85 cm, and 110 cm from each surface. When the drone was hovering less than 10 cm away from the target (i.e., *dead zone*), we recorded its position at 60 Hz for 5 seconds. We repeated this process three times for each target position and computed the mean positional error. Note that we did not consider the rotational error between the target and the drone. When the drone enters the dead zone, the internal PID controller tries to hover the drone inside this zone. The results showed that, on average, the position error was 6.8±1.1 cm. This error happened due to the switching between the positional PID controller and the drone internal PID loop. From Figure 4.20, it is observable that there was no strongly error decreasing pattern when the target distance was increasing. Overall, the minimum positional error was 4 cm while the maximum was 9 cm; the deviation seems to be influenced more by the proximity to the dead zone than to the distance to the obstacle (i.e., higher deviation when the error approaches 10 cm on average). However, we noticed sufficiently stable hovering when the drone was at least 30 cm away from the physical surface. This amount of positional error is acceptable in our application as the virtual augmentation on the drone always follows its real position (tracked drone), not the target location.

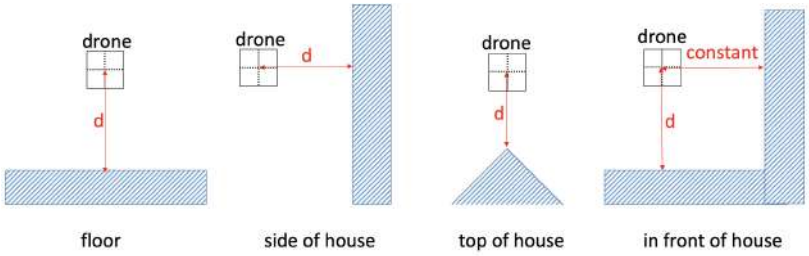

Figure 4.19: Positioning the drone at different target locations with respect to each of the four surfaces. Here, **d** represents distance to the target from the surface.

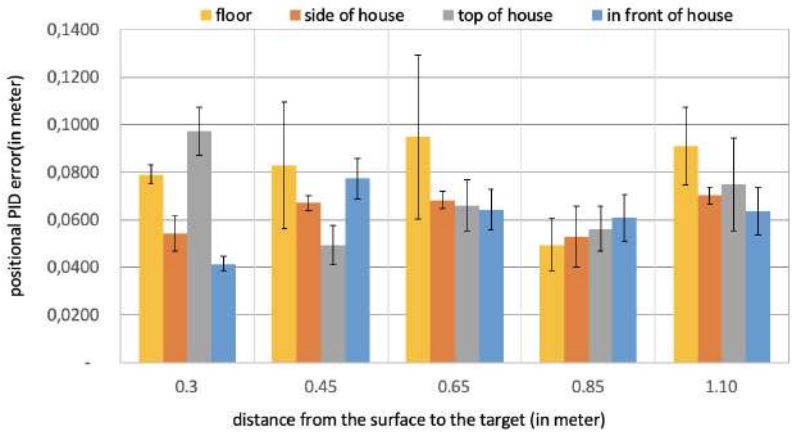

Figure 4.20: The drone positioning error to the target locations for all four surfaces. Overall error is 6.8 ± 1.1cm.

4.6 Limitations

We have shown that combing SAR with a drone opens new opportunities to extend the interaction space. However, even when promising, our approach is not without limitations.

The drone can hover almost perfectly in mid-air (with ±8 cm positional error) when there are no physical objects nearby. This amount of positional error is acceptable as the virtual augmentation on the drone always follows its real position (tracked drone), not the target location. On the other hand, bringing the drone close to the physical objects (e.g., sides or exactly on top of the house) is difficult due to its downwards airflow. Drone hovering was sufficiently stable when it was at least 30 cm away from the physical surface.

The size of the panel attached to the drone is quite small (12 cm x 10 cm) as we restricted ourselves to use a lightweight drone for users' safety. The small size of the drone panel restricts us to project only limited content on it. In order to extend the display surface, it could be possible to combine multiple of these drones to create a bigger surface dynamically. We also envision that there will be improvements in the drone payload capacity and battery life with less noise in the coming years. Blimps might also be an alternative option in this direction, trading speed for projection surface and stability.

The drone does not consider the user's presence while computes a path to reach the goal. In the future, our navigation module should take into account the human position.

Moreover, as we use a front projector, shadows of the user and the drone are inevitable. This could be overcome by using multiple projectors set up [215].

4.7 Conclusion

SAR is strongly linked to the related physical scenes. This makes the user experience with SAR unique, and it provides numerous advantages compared to see-through AR approaches. On the other hand, the physical nature of SAR also induces limitations. We have introduced DroneSAR to overcome some of these limitations. By extending the space on which digital content can be displayed, we have proposed a way to extract the augmentation from the physical constraints. The mid-air drone augmentation is always contextually connected to the physical 3D augmented scene. In our approach, we have explored a set of interactions where users keep immersed in the augmented scene, and they can benefit from additional displays functionalities. This is a proof of concept of how to extend the physical space of SAR using drone augmentation. Once the technology is stable enough, we will conduct a set of user studies to assess the potential and limits of such an extended SAR environment compared to traditional smartphone or tablet-based augmented reality systems.

Conclusion and Future Work

5

In this chapter, we will first revisit our contributions quickly and then present future research directions.

5.1 Revisiting Thesis Contributions

Chapter 3 & Chapter 4 of this dissertation present our two main contributions.

We envision that AR-HMDs are not only designed for manipulating immersive 3D information but also for interacting with 2D content (text, image, video). The major advantage of AR-HMDs over desktop computing is that we can take our workspace with us anywhere we want. In contrast, its traditional input techniques (such as hand tracking, head/eye-gaze, and voice) are cumbersome to use in on-the-go situations. Inspired by previous research, we agree that smartphone has enormous potential as an input device for AR-HMDs when users are not in their desk. In Chapter 3, we investigated how to use this smartphone for selecting text in AR-HMDs. We chose text selection as a use-case because a significant portion of our day-to-day productivity work involves text manipulations, and text selection is often the first step before editing. We efficiently perform this text selection task with a keyboard and mouse/trackpad, but it becomes difficult to achieve in AR-HMDs as its current input techniques hardly support character level precision. To address this issue, we proposed smartphone-based text selection techniques — continuous touch, discrete touch, spatial movement, and raycasting. Next, we compared these techniques in a user study where users have to select text at various granularity levels. Our results suggested that continuous touch (where a smartphone was used as a trackpad) outperformed the other three techniques in terms of task completion time, accuracy, and user preference.

In Chapter 4, we looked into how to embed graphical widgets in spatial augmented reality. Traditional ways of supporting

these widget elements (e.g., via a handheld tracked panel, mobile devices, on the table surface) either limit users' mobility in the augmented space or need to shift their attention frequently. To overcome these issues, we developed *DroneSAR* to provide interactive graphical widgets in SAR like a floating menu in mid-air using projection on a drone-mounted panel. Using a handheld controller, users were able to control the drone position and interact with the scene dynamically. In particular, we presented three ways to embed widgets using a drone in a SAR environment — displaying annotations in mid-air, providing interactive tools, supporting different viewpoints.

5.2 Future Work

In Chapter 3, we focused on the text selection part. In the future, it would be interesting to explore a more global usage scenario such as a text editing interface in AR-HMDs using smartphone-based input where users need to perform other interaction tasks such as text input and commands execution simultaneously. We also need to compare phone-based techniques to other input techniques like hand tracking, head/eye gaze, and voice commands. Furthermore, we only considered standing condition, but it would be interesting to study text selection performance while walking. Note that, there is text readability issue in HMDs due to the blur motion caused by vertical shock while walking [216].

Chapter 4 proposed a novel way to bring a drone into the SAR space. This work can be further extended in a few interesting ways. For example, in our *DroneSAR* system, the size of the drone panel is quite small. Due to this, we are only able to project limited user interface elements on the drone. In this context, it would be interesting to explore spatial menu concepts in the air like virtual shelves [217] and m+pSpaces [218]. Further, we can think about hands-free interaction where users will directly grab the drone to position it [219][210] instead of using a handheld controller. They can also perform direct touch interaction on the drone panels as well as on the augmented surfaces for manipulating virtual contents. Lastly, we need to conduct a user study to investigate the benefits of such drone-projected floating widgets compared to traditional approaches.

We can imagine bringing text selection task in SAR too by projecting textual content on the drone panel (but there is a limitation due to its small panel size) and using the smartphone as an input controller.

5.3 Concluding Remarks

To sum up, some basic tasks (such as text input, text editing, menu selection, interacting with graphical widgets, etc.) are straightforward to do on our desktop but hard to achieve in an immersive AR environment. To overcome these challenges, we have contributed to enhancing the interaction space of the two most commonly used forms of augmented reality. Certainly, proposed interaction techniques are not generic solutions at all. It depends on the application context. Nevertheless, we have just scratched the surface of what is possible with a smartphone and a drone in the context of immersive augmented reality.

Beyond our thesis contribution, we are also working on the commands selection interface for AR-HMDs. While manipulating 3D objects or sketching in AR-HMDs, we often need to issue commands to save the file, change from a brush tool to an eraser tool, changing brush-width, and so on. In a desktop computer, we input commands using 2D UI elements such as toolbars, pull-down menus, pop-up menus, or function keys on a keyboard. On the other hand, designing menus for supporting commands in AR-HMDs is not trivial as these devices have a limited field of view. Displaying menus in an already narrow FOV creates occlusion to the current content. To avoid that, the traditional approach is to use a hand menu* (where the menu is attached to the user's non-dominant hand) or a world-locked floating menu (where the 2D menu is located in the physical space). In both of these menu layouts, users require to explicitly switch their attention from the ongoing tasks to select a menu item. Hence, we asked this question — *is it possible to issue a command rapidly on an AR-HMD without explicitly shifting the user's attention from the primary task?* To address this issue, so far, we have developed a very initial proof-of-concept of a novel head-referenced eyes-free menu layout where menu items (buttons and sliders) are located

* https://docs.microsoft.com/en-us/windows/mixed-reality/design/hand-menu

outside around the FOV. Small dots are shown on the FOV border as visual cues to understand the spatial location of each command. After an initial learning phase, users build a spatial memory of command locations outside the display FOV. This is similar to the way we remember shortcuts for commands on the desktop. Then, users can rapidly select a command using their non-dominant hand with a glance instead of explicit attention switching. The initial implementation is working nicely. Currently, we are planning for a user study to understand its feasibility.

Bibliography

Here are the references in citation order.

[1] Ivan E Sutherland. 'A head-mounted three dimensional display'. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. 1968, pp. 757–764. DOI: https://doi.org/10.1145/1476589.1476686 (cited on pages iv, 1).

[2] Shamsi T Iqbal et al. 'Multitasking with play write, a mobile microproductivity writing tool'. In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 2018, pp. 411–422. DOI: https://doi.org/10.1145/3242587.3242611 (cited on pages iv, 3).

[3] L. H. Lee et al. 'HIBEY: Hide the Keyboard in Augmented Reality'. In: *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*. 2019, pp. 1–10. DOI: 10.1109/PERCOM.2019.8767420 (cited on pages iv, 3, 33).

[4] W. Xu et al. 'Pointing and Selection Methods for Text Entry in Augmented Reality Head Mounted Displays'. In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2019, pp. 279–288. DOI: 10.1109/ISMAR.2019.00026 (cited on pages iv, 3, 33).

[5] John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 'Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion'. In: *ACM Trans. Comput.-Hum. Interact.* 25.6 (Dec. 2018). DOI: 10.1145/3232163 (cited on pages iv, 3, 33).

[6] Duc-Minh Pham and Wolfgang Stuerzlinger. 'HawKEY: Efficient and Versatile Text Entry for Virtual Reality'. In: *25th ACM Symposium on Virtual Reality Software and Technology*. VRST '19. Parramatta, NSW, Australia: Association for Computing Machinery, 2019. DOI: 10.1145/3359996.3364265 (cited on pages iv, 3, 33).

[7] Debjyoti Ghosh et al. 'EYEditor: Towards On-the-Go Heads-Up Text Editing Using Voice and Manual Input'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. DOI: 10.1145/3313831.3376173 (cited on pages iv, v, 3, 34–36).

[8] D. Iwai, T. Yabiki, and K. Sato. 'View Management of Projected Labels on Nonplanar and Textured Surfaces'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.8 (2013), pp. 1415–1424. DOI: 10.1109/TVCG.2012.321 (cited on pages v, 59).

[9] P. Lubos, G. Bruder, and F. Steinicke. 'Analysis of direct selection in head-mounted display environments'. In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. 2014, pp. 11–18. DOI: 10.1109/3DUI.2014.6798834 (cited on pages v, 6, 34).

[10] Li-Wei Chan et al. 'Touching the Void: Direct-Touch Interaction for Intangible Displays'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 2625–2634 (cited on pages v, 34).

[11] Sujin Jang et al. 'Modeling Cumulative Arm Fatigue in Mid-Air Interaction Based on Perceived Exertion and Kinetics of Arm Motion'. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3328–3339. DOI: 10.1145/3025453.3025523 (cited on pages v, 6, 34, 42).

[12] Robert J. K. Jacob. 'What You Look at is What You Get: Eye Movement-Based Interaction Techniques'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. Seattle, Washington, USA: Association for Computing Machinery, 1990, pp. 11–18. DOI: 10.1145/97243.97246 (cited on pages vi, 13, 34).

[13] Colin Ware and Harutune H. Mikaelian. 'An Evaluation of an Eye Tracker as a Device for Computer Input2'. In: *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*. CHI '87. Toronto, Ontario, Canada: Association for Computing Machinery, 1986, pp. 183–188. DOI: 10.1145/29933.275627 (cited on pages vi, 34).

[14] Mikko Kytö et al. 'Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–14 (cited on pages vi, 6, 24, 34).

[15] Yukang Yan et al. 'HeadCross: Exploring Head-Based Crossing Selection on Head-Mounted Displays'. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4.1 (Mar. 2020). DOI: 10.1145/3380983 (cited on pages vi, 13, 34).

[16] Wenge Xu et al. 'DMove: Directional Motion-Based Interaction for Augmented Reality Head-Mounted Displays'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–14. DOI: 10.1145/3290605.3300674 (cited on pages vi, 17, 18, 34).

[17] Yi-Ta Hsieh et al. 'Designing a Willing-to-Use-in-Public Hand Gestural Interaction Technique for Smart Glasses'. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 4203–4215. DOI: 10.1145/2858036.2858436 (cited on pages vi, 6, 34).

[18] Nreal. https://www.nreal.ai/. Accessed: 2020-11-03. 2020 (cited on pages vi, 34).

[19] Wolfgang Büschel et al. 'Investigating Smartphone-Based Pan and Zoom in 3D Data Spaces in Augmented Reality'. In: *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '19. Taipei, Taiwan: Association for Computing Machinery, 2019. DOI: 10.1145/3338286.3340113 (cited on pages vi, 22, 23, 34).

[20] Fengyuan Zhu and Tovi Grossman. 'BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–14. DOI: 10.1145/3313831.3376233 (cited on pages vi, 20, 21, 34).

[21] Jens Grubert et al. '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. Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 3933–3942. DOI: 10.1145/2702123.2702331 (cited on pages vi, 21, 25, 34).

[22] A. Millette and M. J. McGuffin. 'DualCAD: Integrating Augmented Reality with a Desktop GUI and Smartphone Interaction'. In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 2016, pp. 21–26. DOI: 10.1109/ISMAR-Adjunct.2016.0030 (cited on pages vi, 22, 34).

[23] Jie Ren et al. 'Understanding Window Management Interactions in AR Headset + Smartphone Interface'. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–8. DOI: 10.1145/3334480.3382812 (cited on pages vi, 21, 22, 34).

[24] Hyeongmook Lee, Dongchul Kim, and Woontack Woo. 'Graphical menus using a mobile phone for wearable AR systems'. In: *2011 International Symposium on Ubiquitous Virtual Reality*. IEEE. 2011, pp. 55–58 (cited on pages vi, 34).

[25] *Mac keyboard shortcuts*. https://support.apple.com/en-us/HT201236. Accessed: 2020-11-01 (cited on pages vi, 43).

[26] Chris Hoffman. *42+ Text-Editing Keyboard Shortcuts That Work Almost Everywhere*. https://www.howtogeek.com/115664/42-text-editing-keyboard-shortcuts-that-work-almost-everywhere/. Accessed: 2020-11-01 (cited on pages vi, 43).

[27] *iPhone Air Mouse*. http://mobilemouse.com/. Accessed: 2020-11-03 (cited on pages vii, 44).

[28] *Nintendo Wii*. http://wii.com/. Accessed: 2020-11-03 (cited on pages vii, 44).

[29] Doug A. Bowman et al. *3D User Interfaces: Theory and Practice*. USA: Addison Wesley Longman Publishing Co., Inc., 2004 (cited on pages vii, 44).

[30] Mark R. Mine. *Virtual Environment Interaction Techniques*. Tech. rep. USA, 1995 (cited on pages vii, 44).

[31] D. Bandyopadhyay, R. Raskar, and H. Fuchs. 'Dynamic shader lamps : painting on movable objects'. In: *Proceedings IEEE and ACM International Symposium on Augmented Reality*. 2001, pp. 207–216. DOI: 10.1109/ISAR.2001.970539 (cited on pages ix, 6, 27, 32, 67, 68).

[32] John Underkoffler and Hiroshi Ishii. 'Urp: A Luminous-Tangible Workbench for Urban Planning and Design'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1999, pp. 386–393. DOI: 10.1145/302979.303114 (cited on pages ix, 69).

[33] Joan Sol Roo and Martin Hachet. 'One Reality: Augmenting How the Physical World is Experienced by Combining Multiple Mixed Reality Modalities'. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 787–795. DOI: 10.1145/3126594.3126638 (cited on pages x, 31, 32, 69).

[34] Ivan Sutherland. 'The ultimate display'. In: (1965) (cited on page 1).

[35] Feiyu Lu and Doug A Bowman. 'Evaluating the potential of Glanceable AR interfaces for authentic everyday uses'. In: *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2021, pp. 768–777. DOI: 10.1109/VR50410.2021.00104 (cited on page 1).

[36] Leonardo Pavanatto et al. 'Do we still need physical monitors? An evaluation of the usability of AR virtual monitors for productivity work'. In: *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2021, pp. 759–767. DOI: 10.1109/VR50410.2021.00103 (cited on pages 1, 2).

[37] Barrett Ens et al. 'Spatial constancy of surface-embedded layouts across multiple environments'. In: *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. 2015, pp. 65–68. DOI: https://doi.org/10.1145/2788940.2788954 (cited on page 1).

[38] Barrett Ens and Pourang Irani. 'Spatial analytic interfaces: Spatial user interfaces for in situ visual analytics'. In: *IEEE computer graphics and applications* 37.2 (2016), pp. 66–79. DOI: 10.1109/MCG.2016.38 (cited on pages 1, 3).

[39] Joon Hyub Lee et al. 'Projective windows: bringing windows in space to the fingertip'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–8. DOI: https://doi.org/10.1145/3173574.3173792 (cited on page 2).

[40] 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*. 2019, pp. 147–160. DOI: https://doi.org/10.1145/3332165.3347945 (cited on pages 2, 3).

[41] 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*. 2019, pp. 356–366. DOI: https://doi.org/10.1145/3301275.3302278 (cited on pages 2, 3).

[42] Shakiba Davari, Feiyu Lu, and Doug A Bowman. 'Occlusion Management Techniques for Everyday Glanceable AR Interfaces'. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE. 2020, pp. 324–330. DOI: 10.1109/VRW50115.2020.00072 (cited on page 2).

[43] Feiyu Lu et al. 'Glanceable ar: Evaluating information access methods for head-worn augmented reality'. In: *2020 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE. 2020, pp. 930–939. DOI: 10.1109/VR46266.2020.00113 (cited on page 2).

[44] Ashwin Ram and Shengdong Zhao. 'LSVP: Towards Effective On-the-go Video Learning Using Optical Head-Mounted Displays'. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.1 (2021), pp. 1–27. DOI: https://doi.org/10.1145/3448118 (cited on page 2).

[45] Hrvoje Benko, Andrew D. Wilson, and Federico Zannier. 'Dyadic Projected Spatial Augmented Reality'. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, pp. 645–655. DOI: 10.1145/2642918.2647402 (cited on pages 4, 60).

[46] Jun Rekimoto and Masanori Saitoh. 'Augmented surfaces: a spatially continuous work space for hybrid computing environments'. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1999, pp. 378–385. DOI: https://doi.org/10.1145/302979.303113 (cited on pages 4, 5).

[47] Pierre Wellner. 'The DigitalDesk calculator: tangible manipulation on a desk top display'. In: *Proceedings of the 4th annual ACM symposium on User interface software and technology*. 1991, pp. 27–33. DOI: https://doi.org/10.1145/120782.120785 (cited on pages 4, 5).

[48] Pierre Wellner. 'Interacting with paper on the DigitalDesk'. In: *Communications of the ACM* 36.7 (1993), pp. 87–96. DOI: https://doi.org/10.1145/159544.159630 (cited on page 4).

[49] David Holman et al. 'Paper windows: interaction techniques for digital paper'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005, pp. 591–599. DOI: https://doi.org/10.1145/1054972.1055054 (cited on pages 4, 5).

[50] Jeremy Laviole and Martin Hachet. 'Spatial augmented reality for physical drawing'. In: *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012, pp. 9–10. DOI: https://doi.org/10.1145/2380296.2380302 (cited on page 4).

[51] Ramesh Raskar et al. '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*. 1998, pp. 179–188. DOI: https://doi.org/10.1145/280814.280861 (cited on page 4).

[52] Ramesh Raskar et al. 'Shader Lamps: Animating Real Objects With Image-Based Illumination'. In: *Eurographics Workshop on Rendering*. Ed. by S. J. Gortle and K. Myszkowski. The Eurographics Association, 2001. DOI: 10.2312/EGWR/EGWR01/089-101 (cited on pages 4, 5, 27, 59).

[53] Brett R Jones et al. 'IllumiRoom: peripheral projected illusions for interactive experiences'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 869–878. DOI: https://doi.org/10.1145/2470654.2466112 (cited on pages 4, 6).

[54] Brett Jones et al. 'Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units'. In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 2014, pp. 637–644. DOI: https://doi.org/10.1145/2642918.2647383 (cited on pages 4, 5).

[55] Parinya Punpongsanon, Daisuke Iwai, and Kosuke Sato. 'Projection-based visualization of tangential deformation of nonrigid surface by deformation estimation using infrared texture'. In: *Virtual Reality* 19.1 (2015), pp. 45–56 (cited on page 5).

[56] Yi Zhou et al. 'Pmomo: Projection mapping on movable 3D object'. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 781–790. DOI: https://doi.org/10.1145/2858036.2858329 (cited on page 5).

[57] M. R. Marner, B. H. Thomas, and C. Sandor. 'Physical-virtual tools for spatial augmented reality user interfaces'. In: *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. 2009, pp. 205–206. DOI: 10.1109/ISMAR.2009.5336458 (cited on pages 6, 28, 29, 32, 67).

[58] Y. J. Park et al. 'DesignAR: Portable projection-based AR system specialized in interior design'. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017, pp. 2879–2884. DOI: 10.1109/SMC.2017.8123064 (cited on pages 6, 30–32, 67).

[59] *Direct manipulation with hands*. https://docs.microsoft.com/en-us/windows/mixed-reality/direct-manipulation. Accessed: 2020-11-03 (cited on page 9).

[60] *Point and commit with hands*. https://docs.microsoft.com/en-us/windows/mixed-reality/point-and-commit. Accessed: 2020-11-03 (cited on page 9).

[61] Paul Christopher Gloumeau, Wolfgang Stuerzlinger, and JungHyun Han. 'PinNPivot: Object Manipulation using Pins in Immersive Virtual Environments'. In: *IEEE transactions on visualization and computer graphics* (2020). DOI: 10.1109/TVCG.2020.2987834 (cited on pages 9, 10).

[62] Rahul Arora et al. 'Magicalhands: Mid-air hand gestures for animating in vr'. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 2019, pp. 463–477. DOI: https://doi.org/10.1145/3332165.3347942 (cited on pages 9, 11).

[63] Thammathip Piumsomboon et al. 'User-defined gestures for augmented reality'. In: *IFIP Conference on Human-Computer Interaction*. Springer. 2013, pp. 282–299. DOI: https://doi.org/10.1007/978-3-642-40480-1_18 (cited on page 9).

[64] Yukang Yan et al. '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*. 2018, pp. 1–13. DOI: https://doi.org/10.1145/3173574.3173652 (cited on page 10).

[65] Kadek Ananta Satriadi et al. 'Augmented reality map navigation with freehand gestures'. In: *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE. 2019, pp. 593–603. DOI: 10.1109/VR.2019.8798340 (cited on page 10).

[66] Chanho Park et al. 'HandPoseMenu: Hand Posture-Based Virtual Menus for Changing Interaction Mode in 3D Space'. In: *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. 2019, pp. 361–366. DOI: https://doi.org/10.1145/3343055.3360752 (cited on page 10).

[67] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 'Experimental analysis of barehand mid-air mode-switching techniques in virtual reality'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–14. DOI: https://doi.org/10.1145/3290605.3300426 (cited on pages 10, 11).

[68] Wenge Xu et al. 'Pointing and selection methods for text entry in augmented reality head mounted displays'. In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2019, pp. 279–288. DOI: 10.1109/ISMAR.2019.00026 (cited on page 10).

[69] Conor R Foy et al. 'Understanding, Detecting and Mitigating the Effects of Coactivations in Ten-Finger Mid-Air Typing in Virtual Reality'. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–11. DOI: https://doi.org/10.1145/3411764.3445671 (cited on page 10).

[70] Xin Yi et al. 'Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data'. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 539–548. DOI: https://doi.org/10.1145/2807442.2807504 (cited on page 10).

[71] Daniel Schneider et al. 'Accuracy of commodity finger tracking systems for virtual reality head-mounted displays'. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE. 2020, pp. 805–806. DOI: 10.1109/VRW50115.2020.00253 (cited on page 10).

[72] Robert Xiao et al. 'MRTouch: Adding touch input to head-mounted mixed reality'. In: *IEEE transactions on visualization and computer graphics* 24.4 (2018), pp. 1653–1660. DOI: 10.1109/TVCG.2018.2794222 (cited on pages 11, 12).

[73] Chun Yu et al. 'Tap, dwell or gesture? Exploring head-based text entry techniques for HMDs'. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 4479–4488. DOI: https://doi.org/10.1145/3025453.3025964 (cited on page 12).

[74] Anna Maria Feit et al. 'Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design'. In: *Proceedings of the 2017 Chi conference on human factors in computing systems*. 2017, pp. 1118–1130. DOI: https://doi.org/10.1145/3025453.3025599 (cited on page 12).

[75] Difeng Yu et al. 'Depthmove: Leveraging head motions in the depth dimension to interact with virtual reality head-worn displays'. In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2019, pp. 103–114. DOI: 10.1109/ISMAR.2019.00-20 (cited on page 12).

[76] Xueshi Lu et al. 'Depthtext: Leveraging head movements towards the depth dimension for hands-free text entry in mobile virtual reality systems'. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2019, pp. 1060–1061. DOI: 10.1109/VR.2019.8797901 (cited on page 12).

[77] Yukang Yan et al. 'Headgesture: Hands-free input approach leveraging head movements for hmd devices'. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018), pp. 1–23. DOI: https://doi.org/10.1145/3287076 (cited on pages 12, 13).

[78] Yuexing Luo and Daniel Vogel. 'Crossing-based selection with direct touch input'. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2014, pp. 2627–2636. DOI: https://doi.org/10.1145/2556288.2557397 (cited on page 13).

[79] Boris Velichkovsky, Andreas Sprenger, and Pieter Unema. 'Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem"'. In: *Human-Computer Interaction INTERACT'97*. Springer. 1997, pp. 509–516. DOI: https://doi.org/10.1007/978-0-387-35175-9_77 (cited on page 14).

[80] Augusto Esteves et al. 'SmoothMoves: Smooth pursuits head movements for augmented reality'. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 2017, pp. 167–178. DOI: https://doi.org/10.1145/3126594.3126616 (cited on page 14).

[81] Argenis Ramirez Ramirez Gomez et al. 'Gaze+ Hold: Eyes-only Direct Manipulation with Continuous Gaze Modulated by Closure of One Eye'. In: *ACM Symposium on Eye Tracking Research and Applications*. 2021, pp. 1–12. DOI: https://doi.org/10.1145/3448017.3457381 (cited on page 14).

[82] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. 'Gaze gestures or dwell-based interaction?' In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2012, pp. 229–232. DOI: https://doi.org/10.1145/2168556.2168602 (cited on page 14).

[83] Misahael Fernandez, Florian Mathis, and Mohamed Khamis. 'GazeWheels: Comparing Dwell-time Feedback and Methods for Gaze Input'. In: *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*. 2020, pp. 1–6. DOI: https://doi.org/10.1145/3419249.3420122 (cited on page 14).

[84] Aunnoy K Mutasim, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 'Pinch, Click, or Dwell: Comparing Different Selection Techniques for Eye-Gaze-Based Pointing in Virtual Reality'. In: *ACM Symposium on Eye Tracking Research and Applications*. 2021, pp. 1–7. DOI: https://doi.org/10.1145/3448018.3457998 (cited on page 14).

[85] Ludwig Sidenmark et al. 'Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13. DOI: https://doi.org/10.1145/3313831.3376438 (cited on pages 14, 15).

[86] Chang Liu, Jason Orlosky, and Alexander Plopski. 'Eye Gaze-based Object Rotation for Head-mounted Displays'. In: *Symposium on Spatial User Interaction*. 2020, pp. 1–9. DOI: https://doi.org/10.1145/3385959.3418444 (cited on pages 14, 15).

[87] Huidong Bai et al. 'A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing'. In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–13. DOI: https://doi.org/10.1145/3313831.3376550 (cited on page 15).

[88] Kunal Gupta, Gun A Lee, and Mark Billinghurst. 'Do you see what I see? The effect of gaze tracking on task space remote collaboration'. In: *IEEE transactions on visualization and computer graphics* 22.11 (2016), pp. 2413–2422. DOI: 10.1109/TVCG.2016.2593778 (cited on page 15).

[89] Thammathip Piumsomboon et al. 'The effects of sharing awareness cues in collaborative mixed reality'. In: *Frontiers in Robotics and AI* 6 (2019), p. 5. DOI: https://doi.org/10.3389/frobt.2019.00005 (cited on page 15).

[90] Allison Jing et al. 'eyemR-Vis: A Mixed Reality System to Visualise Bi-Directional Gaze Behavioural Cues Between Remote Collaborators'. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–4. DOI: https://doi.org/10.1145/3411763.3451545 (cited on page 15).

[91] 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*. 2011, pp. 441–450. DOI: https://doi.org/10.1145/2047196.2047255 (cited on page 16).

[92] Cheng-Yao Wang et al. 'PalmType: Using palms as keyboards for smart glasses'. In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2015, pp. 153–160. DOI: https://doi.org/10.1145/2785830.2785886 (cited on page 16).

[93] Cheng-Yao Wang et al. 'PalmGesture: Using palms as gesture interfaces for eyes-free input'. In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2015, pp. 217–226. DOI: https://doi.org/10.1145/2785830.2785885 (cited on pages 16, 17).

[94] Sean G Gustafson, Bernhard Rabe, and Patrick M Baudisch. 'Understanding palm-based imaginary interfaces: the role of visual and tactile cues when browsing'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 889–898. DOI: https://doi.org/10.1145/2470654.2466114 (cited on page 16).

[95] Takumi Azai et al. 'Selection and manipulation methods for a menu widget on the human forearm'. In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2017, pp. 357–360. DOI: https://doi.org/10.1145/3027063.3052959 (cited on page 16).

[96] Takumi Azai et al. 'Tap-tap menu: body touching for virtual interactive menus'. In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. 2018, pp. 1–2. DOI: https://doi.org/10.1145/3281505.3281561 (cited on page 17).

[97] Jay Henderson, Jessy Ceha, and Edward Lank. 'STAT: Subtle Typing Around the Thigh for Head-Mounted Displays'. In: *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2020, pp. 1–11. DOI: https://doi.org/10.1145/3379503.3403549 (cited on pages 17, 18).

[98] Florian Müller et al. 'Mind the tap: Assessing foot-taps for interacting with head-mounted displays'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–13. DOI: https://doi.org/10.1145/3290605.3300707 (cited on pages 17, 18).

[99] DoYoung Lee, SooHwan Lee, and Ian Oakley. 'Nailz: Sensing Hand Input with Touch Sensitive Nails'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13. DOI: https://doi.org/10.1145/3313831.3376778 (cited on page 17).

[100] Hsin-Liu Kao et al. 'NailO: fingernails as an input surface'. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 3015–3018. DOI: https://doi.org/10.1145/2702123.2702572 (cited on page 17).

[101] Marcos Serrano, Barrett M Ens, and Pourang P Irani. 'Exploring the use of hand-to-face input for interacting with head-worn displays'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 3181–3190. DOI: https://doi.org/10.1145/2556288.2556984 (cited on page 17).

[102] Yueting Weng et al. 'FaceSight: Enabling Hand-to-Face Gesture Interaction on AR Glasses with a Downward-Facing Camera Vision'. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14. DOI: https://doi.org/10.1145/3411764.3445484 (cited on page 17).

[103] Roman Lissermann et al. 'Earput: Augmenting behind-the-ear devices for ear-based interaction'. In: *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2013, pp. 1323–1328. DOI: https://doi.org/10.1145/2468356.2468592 (cited on page 17).

[104] Yu-Chun Chen et al. 'Exploring User Defined Gestures for Ear-Based Interactions'. In: *Proceedings of the ACM on Human-Computer Interaction* 4.ISS (2020), pp. 1–20. DOI: https://doi.org/10.1145/3427314 (cited on page 17).

[105] Dong-Bach Vo, Eric Lecolinet, and Yves Guiard. 'Belly gestures: body centric gestures on the abdomen'. In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. 2014, pp. 687–696. DOI: https://doi.org/10.1145/2639189.2639210 (cited on page 17).

[106] Duc-Minh Pham and Wolfgang Stuerzlinger. 'Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality'. In: *25th ACM Symposium on Virtual Reality Software and Technology*. 2019, pp. 1–11. DOI: https://doi.org/10.1145/3359996.3364264 (cited on pages 18, 19).

[107] Anil Ufuk Batmaz, Aunnoy K Mutasim, and Wolfgang Stuerzlinger. 'Precision vs. power grip: A comparison of pen grip styles for selection in virtual reality'. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE. 2020, pp. 23–28. DOI: 10.1109/VRW50115.2020.00012 (cited on page 18).

[108] Danilo Gasques et al. 'Pintar: Sketching spatial experiences in augmented reality'. In: *Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion*. 2019, pp. 17–20. DOI: https://doi.org/10.1145/3301019.3325158 (cited on pages 18, 19).

[109] Rahul Arora et al. 'Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–15. DOI: https://doi.org/10.1145/3173574.3173759 (cited on page 18).

[110] Philipp Wacker et al. 'Physical guides: an analysis of 3D sketching performance on physical objects in augmented reality'. In: *Proceedings of the Symposium on Spatial User Interaction*. 2018, pp. 25–35. DOI: https://doi.org/10.1145/3267782.3267788 (cited on page 20).

[111] Bret Jackson, Logan B Caraco, and Zahara M Spilka. 'Arc-Type and Tilt-Type: Pen-based Immersive Text Input for Room-Scale VR'. In: *Symposium on Spatial User Interaction*. 2020, pp. 1–10. DOI: https://doi.org/10.1145/3385959.3418454 (cited on page 20).

[112] E. Normand and M. J. McGuffin. 'Enlarging a Smartphone with AR to Create a Handheld VESAD (Virtually Extended Screen-Aligned Display)'. In: *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2018, pp. 123–133. DOI: 10.1109/ISMAR.2018.00043 (cited on pages 21, 22).

[113] Ricardo Langner et al. 'MARVIS: Combining Mobile Devices and Augmented Reality for Visual Data Analysis'. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–17. DOI: https://doi.org/10.1145/3411764.3445593 (cited on pages 21, 22, 25).

[114] Sebastian Hubenschmid et al. 'STREAM: Exploring the Combination of Spatially-Aware Tablets with Augmented Reality Head-Mounted Displays for Immersive Analytics'. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14. DOI: https://doi.org/10.1145/3411764.3445298 (cited on page 21).

[115] Mohammed Al-Sada et al. 'Input Forager: A User-Driven Interaction Adaptation Approach for Head Worn Displays'. In: *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*. MUM '16. Rovaniemi, Finland: Association for Computing Machinery, 2016, pp. 115–122. DOI: 10.1145/3012709.3012719 (cited on page 21).

[116] Chi-Jung Lee and Hung-Kuo Chu. 'Dual-MR: Interaction with Mixed Reality Using Smartphones'. In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. Tokyo, Japan: Association for Computing Machinery, 2018. DOI: 10.1145/3281505.3281618 (cited on page 21).

[117] Kristoffer Waldow et al. 'An Evaluation of Smartphone-Based Interaction in AR for Constrained Object Manipulation'. In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. Tokyo, Japan: Association for Computing Machinery, 2018. DOI: 10.1145/3281505.3281608 (cited on page 21).

[118] Hyocheol Ro et al. 'AR pointer: Advanced ray-casting interface using laser pointer metaphor for object manipulation in 3D augmented reality environment'. English. In: *Applied Sciences (Switzerland)* 9.15 (Aug. 2019). DOI: 10.3390/app9153078 (cited on page 22).

[119] R. Budhiraja, G. A. Lee, and M. Billinghurst. 'Using a HHD with a HMD for mobile AR interaction'. In: *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2013, pp. 1–6. DOI: 10.1109/ISMAR.2013.6671837 (cited on page 22).

[120] Yuan Chen, Keiko Katsuragawa, and Edward Lank. 'Understanding Viewport- and World-Based Pointing with Everyday Smart Devices in Immersive Augmented Reality'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. DOI: 10.1145/3313831.3376592 (cited on page 22).

[121] Richard A Bolt. '"Put-that-there" Voice and gesture at the graphics interface'. In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 1980, pp. 262–270. DOI: https://doi.org/10.1145/800250.807503 (cited on page 23).

[122] Alex Olwal, Hrvoje Benko, and Steven Feiner. 'Senseshapes: Using statistical geometry for object selection in a multimodal augmented reality'. In: *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* IEEE. 2003, pp. 300–301. DOI: 10.1109/ISMAR.2003.1240730 (cited on page 23).

[123] Adam S Williams, Jason Garcia, and Francisco Ortega. 'Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation'. In: *IEEE Transactions on Visualization and Computer Graphics* 26.12 (2020), pp. 3479–3489. DOI: 10.1109/TVCG.2020.3023566 (cited on page 23).

[124] Ed Kaiser et al. 'Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality'. In: *Proceedings of the 5th international conference on Multimodal interfaces*. 2003, pp. 12–19. DOI: https://doi.org/10.1145/958432.958438 (cited on page 23).

[125] Thammathip Piumsomboon et al. 'Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality'. In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2014, pp. 73–82. DOI: 10.1109/ISMAR.2014.6948411 (cited on page 23).

[126] Di Laura Chen, Ravin Balakrishnan, and Tovi Grossman. 'Disambiguation techniques for freehand object manipulations in virtual reality'. In: *2020 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE. 2020, pp. 285–292. DOI: 10.1109/vr46266.2020.00048 (cited on page 23).

[127] Jiban Adhikary and Keith Vertanen. 'Text Entry in Virtual Environments using Speech and a Midair Keyboard'. In: *IEEE Transactions on Visualization and Computer Graphics* 27.5 (2021), pp. 2648–2658. DOI: 10.1109/TVCG.2021.3067776 (cited on pages 23, 24).

[128] Ken Pfeuffer et al. 'Gaze+ pinch interaction in virtual reality'. In: *Proceedings of the 5th Symposium on Spatial User Interaction*. 2017, pp. 99–108. DOI: https://doi.org/10.1145/3131277.3132180 (cited on page 24).

[129] Wenxin Feng et al. 'HGaze Typing: Head-Gesture Assisted Gaze Typing'. In: *ACM Symposium on Eye Tracking Research and Applications*. 2021, pp. 1–11. DOI: https://doi.org/10.1145/3448017.3457379 (cited on page 24).

[130] Ludwig Sidenmark et al. 'Radi-Eye: Hands-Free Radial Interfaces for 3D Interaction using Gaze-Activated Head-Crossing'. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–11. DOI: https://doi.org/10.1145/3411764.3445697 (cited on page 25).

[131] Ludwig Sidenmark and Hans Gellersen. 'Eye&head: Synergetic eye and head movement for gaze pointing and selection'. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 2019, pp. 1161–1174. DOI: https://doi.org/10.1145/3332165.3347921 (cited on page 25).

[132] Hrvoje Benko et al. 'FoveAR: Combining an Optically See-Through Near-Eye Display with Projector-Based Spatial Augmented Reality'. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, pp. 129–135. DOI: 10.1145/2807442.2807493 (cited on page 25).

[133] Balasaravanan Thoravi Kumaravel et al. 'TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators'. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 182–195. DOI: https://doi.org/10.1145/3379337.3415827 (cited on page 25).

[134] Patrick Reipschläger, Severin Engert, and Raimund Dachselt. 'Augmented Displays: Seamlessly Extending Interactive Surfaces With Head-Mounted Augmented Reality'. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–4. DOI: https://doi.org/10.1145/3334480.3383138 (cited on page 25).

[135] P. Reipschlager, T. Flemisch, and R. Dachselt. 'Personal Augmented Reality for Information Visualization on Large Interactive Displays'. In: vol. 27. 02. Los Alamitos, CA, USA: IEEE Computer Society, Feb. 2021, pp. 1182–1192. DOI: 10.1109/TVCG.2020.3030460 (cited on pages 25, 26).

[136] Jeremy Hartmann, Yen-Ting Yeh, and Daniel Vogel. 'AAR: Augmenting a Wearable Augmented Reality Display with an Actuated Head-Mounted Projector'. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 445–458. DOI: https://doi.org/10.1145/3379337.3415849 (cited on page 26).

[137] Pascal Jansen et al. 'ShARe: Enabling Co-Located Asymmetric Multi-User Interaction for Augmented Reality Head-Mounted Displays'. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 459–471. DOI: https://doi.org/10.1145/3379337.3415843 (cited on page 26).

[138] Chiu-Hsuan Wang et al. 'HMD light: Sharing In-VR experience via head-mounted projector for asymmetric interaction'. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 472–486. DOI: https://doi.org/10.1145/3379337.3415847 (cited on page 26).

[139] Hunter G Hoffman et al. 'Physically touching and tasting virtual objects enhances the realism of virtual experiences'. In: *Virtual Reality* 3.4 (1998), pp. 226–234. DOI: https://doi.org/10.1007/BF01408703 (cited on page 27).

[140] Colin Ware and Jeff Rose. 'Rotating virtual objects with real handles'. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 6.2 (1999), pp. 162–180. DOI: https://doi.org/10.1145/319091.319102 (cited on page 27).

[141] George W Fitzmaurice, Hiroshi Ishii, and William AS Buxton. 'Bricks: laying the foundations for graspable user interfaces'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995, pp. 442–449. DOI: https://doi.org/10.1145/223904.223964 (cited on page 27).

[142] Brett R Jones et al. 'Build your world and play in it: Interacting with surface particles on complex objects'. In: *2010 IEEE international symposium on mixed and augmented reality*. IEEE. 2010, pp. 165–174. DOI: 10.1109/ISMAR.2010.5643566 (cited on pages 27, 28, 32).

[143] Steven Henderson and Steven Feiner. 'Opportunistic tangible user interfaces for augmented reality'. In: *IEEE Transactions on Visualization and Computer Graphics* 16.1 (2009), pp. 4–16. DOI: 10.1109/TVCG.2009.91 (cited on pages 28, 32).

[144] Michael R Marner et al. 'Spatial user interfaces for large-scale projector-based augmented reality'. In: *IEEE computer graphics and applications* 34.6 (2014), pp. 74–82. DOI: 10.1109/MCG.2014.117 (cited on pages 29, 32).

[145] Susanne Schmidt et al. 'Floor-Projected Guidance Cues for Collaborative Exploration of Spatial Augmented Reality Setups'. In: *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ISS '18. Tokyo, Japan: Association for Computing Machinery, 2018, pp. 279–289. DOI: 10.1145/3279778.3279806 (cited on pages 29, 30, 32, 67).

[146] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 'MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 199–208. DOI: 10.1145/2207676.2207704 (cited on pages 29, 32).

[147] Jeremy Hartmann and Daniel Vogel. 'An Evaluation of Mobile Phone Pointing in Spatial Augmented Reality'. In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–6. DOI: https://doi.org/10.1145/3170427.3188535 (cited on pages 29, 30, 32).

[148] Renaud Gervais, Jérémy Frey, and Martin Hachet. 'Pointing in spatial augmented reality from 2D pointing devices'. In: *IFIP Conference on Human-Computer Interaction*. Springer. 2015, pp. 381–389. DOI: https://doi.org/10.1007/978-3-319-22723-8_30 (cited on pages 30–32).

[149] J. S. Roo and M. Hachet. 'Towards a hybrid space combining Spatial Augmented Reality and virtual reality'. In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. 2017, pp. 195–198. DOI: 10.1109/3DUI.2017.7893339 (cited on pages 31, 32).

[150] Stuart K. Card, William K. English, and Betty J. Burr. 'Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT'. In: *Ergonomics* 21.8 (1978), pp. 601–613. DOI: 10.1080/00140137808931762 (cited on page 34).

[151] Ishan Chatterjee, Robert Xiao, and Chris Harrison. 'Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions'. In: ICMI '15. Seattle, Washington, USA: Association for Computing Machinery, 2015, pp. 131–138. DOI: 10.1145/2818346.2820752 (cited on page 34).

[152] Shyamli Sindhwani, Christof Lutteroth, and Gerald Weber. 'ReType: Quick Text Editing with Keyboard and Gaze'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–13. DOI: 10.1145/3290605.3300433 (cited on page 34).

[153] Alix Goguey, Sylvain Malacria, and Carl Gutwin. 'Improving Discoverability and Expert Performance in Force-Sensitive Text Selection for Touch Devices with Mode Gauges'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–12. DOI: 10.1145/3173574.3174051 (cited on pages 34, 39).

[154] Magic Leap. *Magic Leap Handheld Controller*. https://developer.magicleap.com/en-us/learn/guides/design-magic-leap-one-control. Accessed: 2020-11-01. 2020 (cited on page 34).

[155] L. Lee et al. 'One-thumb Text Acquisition on Force-assisted Miniature Interfaces for Mobile Headsets'. In: *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2020, pp. 1–10. DOI: 10.1109/PerCom45495.2020.9127378 (cited on page 35).

[156] Bryan Campbell et al. 'Fitts' Law Predictions with an Alternative Pointing Device (Wiimote(R))'. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 52 (Sept. 2008). DOI: 10.1177/154193120805201904 (cited on pages 36, 56).

[157] Daniel Vogel and Ravin Balakrishnan. 'Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays'. In: *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. UIST '05. Seattle, WA, USA: Association for Computing Machinery, 2005, pp. 33–42. DOI: 10.1145/1095034.1095041 (cited on pages 36, 37, 56).

[158] Matthias Baldauf et al. 'From Touchpad to Smart Lens'. In: *International Journal of Mobile Human Computer Interaction* 5 (Aug. 2015), pp. 1–20. DOI: 10.4018/jmhci.2013040101 (cited on pages 36, 37, 42).

[159] Mohit Jain, Andy Cockburn, and Sriganesh Madhvanath. 'Comparison of Phone-Based Distal Pointing Techniques for Point-Select Tasks'. In: *14th International Conference on Human-Computer Interaction (INTERACT)*. Ed. by Paula Kotzé et al. Vol. LNCS-8118. Human-Computer Interaction – INTERACT 2013 Part II. Part 15: Mobile Interaction Design. Cape Town, South Africa: Springer, Sept. 2013, pp. 714–721. DOI: 10.1007/978-3-642-40480-1_49 (cited on pages 37, 42, 56).

[160] Shaishav Siddhpuria et al. 'Pointing at a Distance with Everyday Smart Devices'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–11. DOI: https://doi.org/10.1145/3173574.3173747 (cited on pages 37, 56).

[161] Mathieu Nancel et al. 'High-Precision Pointing on Large Wall Displays Using Small Handheld Devices'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, pp. 831–840. DOI: 10.1145/2470654.2470773 (cited on pages 37, 42, 46).

[162] Sebastian Boring et al. 'The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction'. In: *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '12. San Francisco, California, USA: Association for Computing Machinery, 2012, pp. 39–48. DOI: 10.1145/2371574.2371582 (cited on page 37).

[163] Weiyang Huan, Huawei Tu, and Zhuying Li. 'Enabling Finger Pointing Based Text Selection on Touchscreen Mobile Devices'. In: *Proceedings of the Seventh International Symposium of Chinese CHI*. Chinese CHI '19. Xiamen, China: Association for Computing Machinery, 2019, pp. 93–96. DOI: 10.1145/3332169.3332172 (cited on pages 37, 38).

[164] Vittorio Fuccella, Poika Isokoski, and Benoit Martin. 'Gestures and Widgets: Performance in Text Editing on Multi-Touch Capable Mobile Devices'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, pp. 2785–2794. DOI: 10.1145/2470654.2481385 (cited on page 37).

[165] Mingrui Zhang and Jacob O. Wobbrock. 'Gedit: Keyboard Gestures for Mobile Text Editing'. In: *Proceedings of Graphics Interface 2020*. GI 2020. University of Toronto: Canadian Human-Computer Communications Society, 2020, pp. 470–473. DOI: 10.20380/GI2020.47 (cited on pages 37, 38).

[166] Toshiyuki Ando et al. 'Press & Tilt: One-Handed Text Selection and Command Execution on Smartphone'. In: *Proceedings of the 30th Australian Conference on Computer-Human Interaction*. OzCHI '18. Melbourne, Australia: Association for Computing Machinery, 2018, pp. 401–405. DOI: https://doi.org/10.1145/3292147.3292178 (cited on pages 38, 39).

[167] Toshiyuki Ando et al. 'One-Handed Rapid Text Selection and Command Execution Method for Smartphones'. In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI EA '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–6. DOI: 10.1145/3290607.3312850 (cited on pages 38, 39).

[168] Chen Chen et al. 'BezelCopy: An Efficient Cross-Application Copy-Paste Technique for Touchscreen Smartphones'. In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. AVI '14. Como, Italy: Association for Computing Machinery, 2014, pp. 185–192. DOI: 10.1145/2598153.2598162 (cited on page 38).

[169] Huy Viet Le et al. 'Shortcut Gestures for Mobile Text Editing on Fully Touch Sensitive Smartphones'. In: *ACM Trans. Comput.-Hum. Interact.* 27.5 (Aug. 2020). DOI: 10.1145/3396233 (cited on page 39).

[170] Radiah Rivu et al. 'Gaze'N'Touch: Enhancing Text Selection on Mobile Devices Using Gaze'. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–8. DOI: 10.1145/3334480.3382802 (cited on pages 39, 40).

[171] Alexander Keith Eady and Audrey Girouard. 'Caret Manipulation Using Deformable Input in Mobile Devices'. In: *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '15. Stanford, California, USA: Association for Computing Machinery, 2015, pp. 587–591. DOI: 10.1145/2677199.2687916 (cited on page 39).

[172] Andrea Leganchuk, Shumin Zhai, and William Buxton. 'Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study'. In: *ACM Trans. Comput.-Hum. Interact.* 5.4 (Dec. 1998), pp. 326–359. DOI: 10.1145/300520.300522 (cited on page 41).

[173] W. Buxton and B. Myers. 'A Study in Two-Handed Input'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '86. Boston, Massachusetts, USA: Association for Computing Machinery, 1986, pp. 321–326. DOI: 10.1145/22627.22390 (cited on page 41).

[174] Amy K Karlson, Benjamin B Bederson, and J Contreras-Vidal. 'Understanding single-handed mobile device interaction'. In: *Handbook of research on user interface design and evaluation for mobile technology* 1 (2006), pp. 86–101 (cited on page 41).

[175] Alexander Ng, Stephen A Brewster, and John Williamson. 'The impact of encumbrance on mobile interactions'. In: *IFIP Conference on Human-Computer Interaction*. Springer. 2013, pp. 92–109 (cited on page 41).

[176] J. L. Gabbard, D. G. Mehra, and J. E. Swan. 'Effects of AR Display Context Switching and Focal Distance Switching on Human Performance'. In: *IEEE Transactions on Visualization and Computer Graphics* 25.6 (2019), pp. 2228–2241. DOI: 10.1109/TVCG.2018.2832633 (cited on page 41).

[177] Andrew Bragdon et al. 'Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, pp. 403–412. DOI: 10.1145/1978942.1979000 (cited on page 41).

[178] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 'Gunslinger: Subtle Arms-down Mid-air Interaction'. In: *ACM UIST 2015 - 28th Annual ACM Symposium on User Interface Software & Technology*. Charlotte, United States, Nov. 2015, pp. 63–71. DOI: 10.1145/1095034.1095041 (cited on page 42).

[179] Géry Casiez et al. 'RubberEdge: Reducing Clutching by Combining Position and Rate Control with Elastic Feedback'. In: *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST '07. Newport, Rhode Island, USA: Association for Computing Machinery, 2007, pp. 129–138. DOI: 10.1145/1294211.1294234 (cited on page 43).

[180] Géry Casiez, Nicolas Roussel, and Daniel Vogel. '1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems'. In: *CHI'12, the 30th Conference on Human Factors in Computing Systems*. Austin, United States: ACM, May 2012, pp. 2527–2530. DOI: 10.1145/2207676.2208639 (cited on page 47).

[181] Dennis Wolf et al. 'Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–10. DOI: 10.1145/3313831.3376876 (cited on pages 47, 51).

[182] Sandra G Hart. 'NASA-task load index (NASA-TLX); 20 years later'. In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 50. 9. Sage publications Sage CA: Los Angeles, CA. 2006, pp. 904–908 (cited on page 50).

[183] Christian Tonn et al. 'Spatial Augmented Reality for Architecture — Designing and Planning with and within Existing Buildings'. In: *International Journal of Architectural Computing* 6.1 (2008), pp. 41–58. DOI: 10.1260/147807708784640126 (cited on page 59).

[184] Jeremy Laviole et al. 'Nectar: Multi-User Spatial Augmented Reality for Everyone: Three Live Demonstrations of Educative Applications'. In: *Proceedings of the Virtual Reality International Conference - Laval Virtual*. VRIC '18. Laval, France: Association for Computing Machinery, 2018. DOI: 10.1145/3234253.3234317 (cited on page 59).

[185] Brett Ridel et al. 'The revealing flashlight: Interactive spatial augmented reality for detail exploration of cultural heritage artifacts'. In: *Journal on Computing and Cultural Heritage (JOCCH)* 7.2 (2014), pp. 1–18. DOI: https://doi.org/10.1145/2611376 (cited on page 59).

[186] Hajime Kajita, Naoya Koizumi, and Takeshi Naemura. 'SkyAnchor: Optical Design for Anchoring Mid-Air Images onto Physical Objects'. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, pp. 415–423. DOI: 10.1145/2984511.2984589 (cited on pages 60, 62).

[187] Diego Martinez Plasencia et al. 'Through the Combining Glass'. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, pp. 341–350. DOI: 10.1145/2642918.2647351 (cited on page 60).

[188] Asuka Yagi et al. '360-degree fog projection interactive display'. In: *SIGGRAPH Asia 2011 Emerging Technologies*. 2011, pp. 1–1. DOI: https://doi.org/10.1145/2073370.2073388 (cited on pages 61, 62).

[189] Diego Martinez Plasencia, Edward Joyce, and Sriram Subramanian. 'MisTable: reach-through personal screens for tabletops'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 3493–3502. DOI: https://doi.org/10.1145/2556288.2557325 (cited on page 61).

[190] Yutaka Tokuda et al. 'MistForm: Adaptive shape changing fog screens'. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 4383–4395. DOI: https://doi.org/10.1145/3025453.3025608 (cited on page 61).

[191] Peter C Barnum, Srinivasa G Narasimhan, and Takeo Kanade. 'A multi-layered display with water drops'. In: *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–7. DOI: https://doi.org/10.1145/1833349.1778813 (cited on page 62).

[192] Yoichi Ochiai, Takayuki Hoshi, and Jun Rekimoto. 'Pixie dust: graphics generated by levitated and animated objects in computational acoustic-potential field'. In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), pp. 1–13. DOI: https://doi.org/10.1145/2601097.2601118 (cited on pages 62, 63).

[193] Hideo Saito et al. 'Laser-plasma scanning 3D display for putting digital contents in free space'. In: *Stereoscopic Displays and Applications XIX*. Vol. 6803. International Society for Optics and Photonics. 2008, p. 680309. DOI: https://doi.org/10.1117/12.768068 (cited on page 62).

[194] Yoichi Ochiai et al. 'Fairy lights in femtoseconds: aerial and volumetric graphics rendered by focused femtosecond laser combined with computational holographic fields'. In: *ACM Transactions on Graphics (TOG)* 35.2 (2016), pp. 1–14. DOI: https://doi.org/10.1145/2850414 (cited on page 62).

[195] Hanyuool Kim et al. 'MARIO: Mid-air Augmented Reality Interaction with Objects'. In: *Entertainment Computing* 5.4 (2014), pp. 233–241. DOI: https://doi.org/10.1016/j.entcom.2014.10.008 (cited on page 62).

[196] Jürgen Scheible et al. 'Displaydrone: A Flying Robot Based Interactive Display'. In: *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*. PerDis '13. Mountain View, California: Association for Computing Machinery, 2013, pp. 49–54. DOI: 10.1145/2491568.2491580 (cited on pages 63, 64).

[197] Pascal Knierim et al. 'Quadcopter-Projected In-Situ Navigation Cues for Improved Location Awareness'. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–6 (cited on pages 63, 64).

[198] Marius Hoggenmueller and Martin Tomitsch. 'Enhancing Pedestrian Safety through In-Situ Projections: A Hyperreal Design Approach'. In: *Proceedings of the 8th ACM International Symposium on Pervasive Displays*. PerDis '19. Palermo, Italy: Association for Computing Machinery, 2019. DOI: 10.1145/3321335.3329682 (cited on page 63).

[199] Anke M. Brock et al. 'FlyMap: Interacting with Maps Projected from a Drone'. In: *Proceedings of the 7th ACM International Symposium on Pervasive Displays*. PerDis '18. Munich, Germany: Association for Computing Machinery, 2018. DOI: 10.1145/3205873.3205877 (cited on page 63).

[200] Mikhail Matrosov, Olga Volkova, and Dzmitry Tsetserukou. 'LightAir: A Novel System for Tangible Communication with Quadcopters Using Foot Gestures and Projected Image'. In: *ACM SIGGRAPH 2016 Emerging Technologies*. SIGGRAPH '16. Anaheim, California: Association for Computing Machinery, 2016. DOI: 10.1145/2929464.2932429 (cited on pages 63, 64).

[201] J. R. Cauchard et al. 'Drone.io: A Gestural and Visual Interface for Human-Drone Interaction'. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2019, pp. 153–162. DOI: 10.1109/HRI.2019.8673011 (cited on pages 63, 64).

[202] Stefan Schneegass et al. 'Midair Displays: Concept and First Experiences with Free-Floating Pervasive Displays'. In: *Proceedings of The International Symposium on Pervasive Displays*. PerDis '14. Copenhagen, Denmark: Association for Computing Machinery, 2014, pp. 27–31. DOI: 10.1145/2611009.2611013 (cited on page 63).

[203] Hiroki Nozaki. 'Flying Display: A Movable Display Pairing Projector and Screen in the Air'. In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 909–914. DOI: 10.1145/2559206.2579410 (cited on page 63).

[204] Ippei Suzuki et al. 'Gushed Light Field: Design Method for Aerosol-Based Fog Display'. In: *SIGGRAPH ASIA 2016 Posters*. SA '16. Macau: Association for Computing Machinery, 2016. DOI: 10.1145/3005274.3005295 (cited on page 64).

[205] Wataru Yamada et al. 'ISphere: Self-Luminous Spherical Drone Display'. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 635–643. DOI: 10.1145/3126594.3126631 (cited on pages 64, 65).

[206] S. Toyohara, S. Miyafuji, and H. Koike. '[POSTER] ARial Texture: Dynamic Projection Mapping on Drone Propellers'. In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 2017, pp. 206–211. DOI: 10.1109/ISMAR-Adjunct.2017.68 (cited on page 65).

[207] Xujing Zhang et al. 'LightBee: A Self-Levitating Light Field Display for Hologrammatic Telepresence'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–10. DOI: 10.1145/3290605.3300242 (cited on page 65).

[208] Hiroaki Tobita, Shigeaki Maruyama, and Takuya Kuzi. 'Floating Avatar: Telepresence System Using Blimps for Communication and Entertainment'. In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, pp. 541–550. DOI: 10.1145/1979742.1979625 (cited on page 65).

[209] S. Dent. https://www.engadget.com/2017-02-06-intel-drones-form-us-flag-for-lady-gagas-halftime-show.html. Last accessed: 2021-02-15 (cited on page 65).

[210] Antonio Gomes et al. 'BitDrones: Towards Using 3D Nanocopter Displays as Interactive Self-Levitating Programmable Matter'. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 770–780. DOI: 10.1145/2858036.2858519 (cited on pages 65, 66, 80).

[211] Ramesh Raskar and Kok-Lim Low. 'Interacting with Spatially Augmented Reality'. In: *Proceedings of the 1st International Conference on Computer Graphics, Virtual Reality and Visualisation*. AFRIGRAPH '01. Camps Bay, Cape Town, South Africa: Association for Computing Machinery, 2001, pp. 101–108. DOI: 10.1145/513867.513889 (cited on page 66).

[212] K Somani Arun, Thomas S Huang, and Steven D Blostein. 'Least-squares fitting of two 3-D point sets'. In: *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987), pp. 698–700. DOI: 10.1109/TPAMI.1987.4767965 (cited on page 71).

[213] Nghia Ho. *Finding optimal rotation and translation between corresponding 3D points*. http://nghiaho.com/?page_id=671. Last accessed: 17-April-2021 (cited on page 71).

[214] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 'A formal basis for the heuristic determination of minimum cost paths'. In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107. DOI: 10.1109/TSSC.1968.300136 (cited on page 72).

[215] R. Sukthankar, Tat-Jen Cham, and G. Sukthankar. 'Dynamic shadow elimination for multi-projector displays'. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. 2001, pp. II–II. DOI: 10.1109/CVPR.2001.990943 (cited on page 77).

[216] Shogo Fukushima, Takeo Hamada, and Ari Hautasaari. 'Comparing World and Screen Coordinate Systems in Optical See-Through Head-Mounted Displays for Text Readability while Walking'. In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2020, pp. 649–658. DOI: 10.1109/ISMAR50242.2020.00093 (cited on page 80).

[217] Frank Chun Yat Li, David Dearman, and Khai N. Truong. 'Virtual Shelves: Interactions with Orientation Aware Devices'. In: UIST '09. Victoria, BC, Canada: Association for Computing Machinery, 2009, pp. 125–128. DOI: 10.1145/1622176.1622200 (cited on page 80).

[218] Jessica Cauchard et al. '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. San Francisco, California, USA: Association for Computing Machinery, 2012, pp. 171–180. DOI: 10.1145/2371574.2371601 (cited on page 80).

[219] Pascal Knierim et al. 'Flyables: Exploring 3D Interaction Spaces for Levitating Tangibles'. In: *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '18. Stockholm, Sweden: Association for Computing Machinery, 2018, pp. 329–336. DOI: 10.1145/3173225.3173273 (cited on page 80).