HandyNotes: using the hands to create semantic representations of contextually aware real-world objects

Clément Quere, Aline Menin, Raphaël Julien, Hui-Yin Wu, Marco Winckler

HandyNotes: using the hands to create semantic representations of contextually aware real-world objects

Clément Quere[*], Aline Menin[†], Raphaël Julien[‡]
Université Côte d'Azur,
CNRS, Inria, I3S
Sophia-Antipolis, France

Hui-Yin Wu[§]
Inria, Université Côte d'Azur
Sophia-Antipolis, France

Marco Winckler[¶]
Université Côte d'Azur,
CNRS, Inria, I3S
Sophia-Antipolis, France

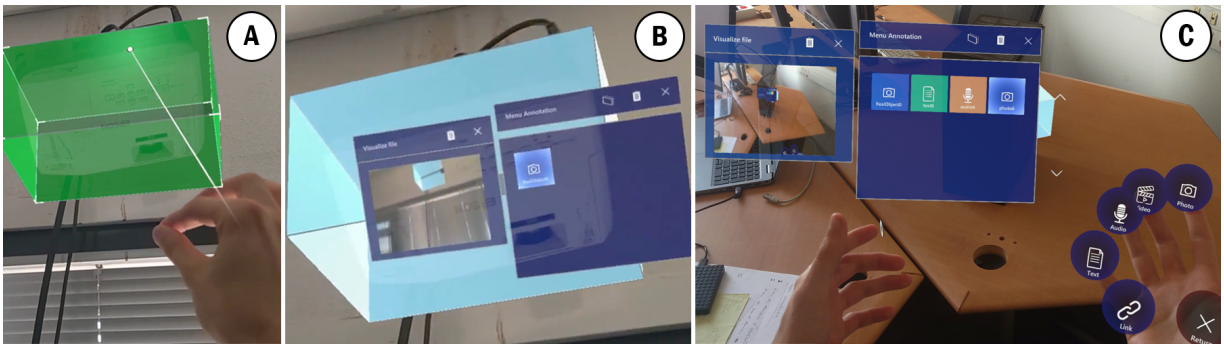

Figure 1: Overview of the HandyNotes system. (a) Creation and manipulation of a bounding box determining the annotation target. (b) Annotation body associated with a bounding box. (c) Visualization of the object's annotations and the Hand-Menu System.

Abstract

This paper uses Mixed Reality (MR) technologies to provide a seamless integration of digital information in physical environments through human-made annotations. Creating digital annotations of physical objects evokes many challenges for performing (simple) tasks such as adding digital notes and connecting them to real-world objects. For that, we have developed an MR system using the Microsoft HoloLens2 to create semantic representations of contextually-aware real-world objects while interacting with holographic virtual objects. User interaction is enhanced with use of fingers as placeholders for menu items. We demonstrate our approach through two real-world scenarios. We also discuss the challenges for using MR technologies.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Human-centered computing—Human computer interaction (HCI)—Interaction techniques—Gestural input; Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Graphical user interfaces;

1 Introduction

Augmented Reality (AR) and Mixed Reality (MR) technology has been sought as a means to make the transition from virtual to physical environments easier for users, by allowing them to engage with holographic virtual objects that are contextually aware of their real-world environment [29]. Nonetheless, appropriate strategies and transition techniques are necessary in different usage contexts of the applications and the user tasks [14, 24]. We observe that tasks such as digital annotation of physical objects still evoke several design challenges to support optimal user experience.

Annotations enable the interaction between separate pieces of information to fulfill various functions: providing descriptions (contextualizing data, citing sources, etc.), offering assessments (identifying quality issues, posing questions or concerns, etc.), or a combination of these purposes [40]. In our daily lives, we routinely create annotations to summarize and emphasize critical elements within written content or to insert reminders, translations, clarifications, or messages for others in collaborative documents. While physical annotation often require pen and paper, any tangible object in our surroundings can serve as a canvas for digital annotations (e.g. by adding a post-it note to a light switch for explanatory purposes) [22]. Thus, MR technologies play a main role by facilitating the virtualization of annotations and seamless interaction with the real world through the use of head-mounted displays (HMDs).

The annotating of physical spaces is very relevant for many application domains. For instance, in real estate, MR-driven space annotations offer customers immersive virtual experiences where they can envision themselves within potential future homes [41]. In healthcare, medical practitioners [17] leverage space annotations to bolster the confidence of patients recovering from trauma or grappling with Alzheimer's disease, providing recognition and comprehension exercises [44]. In entertainment, across artistic endeavors [3] such as cinema, theater, and video games [11], annotations serve as invaluable tools for directors, facilitating the effective communication of their creative vision to their collaborative teams.

This paper explores the capacity of MR to create semantic representations of the real world through human-made annotations. This task hinges upon diverse components, many of which have been thoroughly scrutinized in prior literature, notably within the realms of Virtual Reality (VR) and AR [32]. Thus, there exists a need to develop user-friendly interaction methods with these technologies. The goal is to allow users to navigate and interact with their physical environment in a manner that feels intuitive, effortless, and transparent. Conventional interaction paradigms, such as menus and textual input, may not suffice to fully harness the potential of MR. In the literature, myriad MR paradigms have been explored, encompassing air gestures, voice commands, eye tracking, hand controllers, haptic

[*]e-mail: clement.quere@inria.fr
[†]e-mail: aline.menin@inria.fr
[‡]e-mail: rjulien@digitec.fr
[§]e-mail: hui-yin.wu@inria.fr
[¶]e-mail: marco.winckler@inria.fr

feedback, among others. In this paper, we contribute to the ongoing advancement of MR technology and its applications, particularly in facilitating more meaningful and natural interactions with the real world by addressing four aspects:

- **The design and development of an annotation system**: we present the system *HandyNotes* that empowers users to enrich real-world objects with semantic information. It leverages MR headsets and multiple input channels (e.g. audio, air gestures, and text), enabling seamless annotations of physical environments.
- **Extending the hand menu system interaction paradigm**: We build upon the concept of hand menu systems, which are intuitively attached to the user's hand or wrist, ensuring easy access to tools and options without obstructing the user's field of view.
- **Asynchronous note-taking process**: We offer an asynchronous note-taking process that enables users to share their annotated world.
- **Lessons learned**: We share our experiences in overcoming challenges encountered during the design and development of *HandyNotes* through rational design.

The remainder of the paper is organized as follows. Section 2 summarizes previous works on MR-based annotation systems and MR interaction paradigms. Section 3 describes the annotation tasks and design alternatives. Section 4 presents our system and the challenges encountered during the development. Section 5 illustrates through use cases the usage and usefulness of our tool. Section 6 discusses the limitations at the light of findings issued from a user trial. Lastly, section 7 presents the conclusions and future work.

2 Related Work

Hereafter, we summarize previous contributions in terms of MR interaction paradigms and annotations systems.

2.1 Mixed reality interaction paradigms

Hand gestures MR headsets use gesture recognition algorithms to detect the position of fingers in the virtual world, enabling the user to interact with virtual objects. Hand gesture recognition is a meticulous process as it depends on the relative position of small joints of the hand. Although numerous gestures can be performed by adjusting the position of hand joints [35], this narrows the gap between different gestures, which forces users to perfectly perform the pre-defined gestures [36]. In the meantime, another issue lies in the accuracy of gesture recognition. A threshold in the position of hand joints is necessary in order to recognize hand gestures [37], which hinders their recognition with current technology [26]. Furthermore, long usage time and repeating the same gestures can result in fatigue, which could be slightly reduced when combining gestures with other interaction paradigms, such as gaze and speech [43]. Williams et al. [45] summarized a handful of gestures that are frequently used in MR environments supporting most of the necessary tasks, described as follows. The *pinch* gesture is commonly employed to adjust the position or manipulate an object (e.g. bring a window into the field of view). The *Grab* gesture serve mainly to move objects in the virtual space by dragging and dropping objects. The *Palm up* gesture is usually used to reveal an object or a menu on the top of the hand [8, 34]. The *Index* gesture assist the interaction with digital surfaces, such as clicking on a button or touching the screen, which has been shown to be an effective interaction technique [7]. Amores et al. [2] uses hand gestures to support communication between expert and apprentice within a remote collaboration system.

Hand ray Physical limitations such as the length of a person's arms limits the interaction with distant objects. In this context, the hand ray technique uses a line segment stretching from the user's position (e.g. the hand) to a distant user interface or 3D object [13]. This technique is often combined with the *Index* gesture to support object selection, and *Grab* gestures to support distant object manipulation through actions such as translation, rotation and scaling [8, 25]. Bao et al. [5] combined hand ray and gestures to reach hidden objects in the virtual world.

Gaze Our vision is faster and more intuitive than gestures, making it a good interaction alternative to alleviate arm fatigue caused by the aforementioned interaction paradigms. MR devices often feature eye tracking, which after appropriate calibration supports accurate interactions by adapting to attention and eye speed of the user. Gaze has been used in the literature to support object selection, which can be problematic as the user may accidentally select an object by staring too long at it [42]. Weibel et al. [44] used gaze to capture and keep a record of the user's actions at different moments in time.

Voice commands Voice systems offer superior control over the scene by supporting hands-free interaction. Actions can be defined as command lines (i.e. select, remove, turn off, etc) [43], which users must nevertheless remember and perfectly articulate them.

Hand-Menu Systems Sasaki et al. [34] presented the Hand-Menu System as an instant, place-independent, and suitable input interface for wearable computers. In these systems, when the user looks at their open hands, the menu is revealed, allowing them to perform commands by clicking on a menu icon that appears on the fingertips of the open hand. Azai et al. [4] slightly modified the approach by placing the menu next to the user's hand, which has been shown to reduce the number of parasitic interactions caused by tracking errors due to overlapping of menu and selection hands. Both solutions require the user to keep an open hand, which causes fatigue and handicaps the one-handed user. Pfeuffeur et al. [31] combined menu-hand systems, gaze and pinch gestures to support interaction. In this approach, the user opens their hand to reveal the menu, then looks at the command they wish to select and pinches with the same hand to perform the chosen command.

2.2 Annotations in Mixed Reality

Hereafter, we summarize in the Table 1 summarizes the features of existing MR annotation systems and we compared them to our approach. As we shall see, annotation tasks on MR require: calibration of Virtual Environment (VE), target and body creation, body content, target-body association, and visualization of annotations, which are extensively discussed in Section 3. Borhani et al. [10] explored the usage of asynchronous annotation processes to support adding comments, remarks, or suggestions directly in the MR environment, which can be visually explored by other collaborators at a later time; users can filter out uninteresting annotations to reduce visual clutter. Teo et al. [39] explored the concept of drawing annotations through ray casting, which allows a certain precision to communication. Chang et al. [12] explored mid-air gesture interaction to annotate an object by drawing circles and arrows. These shapes are then replaced by smoother ones, a recurrent approach on 2D frames, which is relevant for annotation [12, 13, 19]. Wong et al. [47] proposed an image annotation system, in which objects present in the image are automatically detected and overlaid by an interactive bounding box, which users can then click and associate tags. Previous works also support collaboration within MR annotation systems. In this context, Herbert et al. [23] suggested that annotated objects should be highlighted to inform remote users that an object is being annotated. For safety reasons, Elvezio et al. [30] proposed that the object should be duplicated to enable experts and local users to perform actions on the same object without errors or loss of information. Chantziaras et al [13] used spatial mesh reconstruction of the environment based

Table 1: Overview of existing MR annotations systems compared to our approach in terms of according to the provided support to the annotation task (see Section 3).

Ref.	Calibration of VE	Target creation	Body creation	Body content	Target-Body association	Visualization
[10]	None	3D objects	Unknown	Textual input, audio & video, picture	Target first, then body	As it is
[12]	Unknown	None	Mid-air drawing	3D objects	Random	As it is
[13]	Spatial mapping	Mesh selection	Text typing, video recording	Textual input, audio & video, picture	Random	As it is
[30]	None	3D objects	3D object placement	3D objects	Target first, then body	Graph, As it is
[23]	Marker-based referential	None	Physical keyboard	Textual input	Target first, then body	Graph, As it is
[39]	None	Hand ray	Mid-air drawing	Mid-air drawings, audio & video	Target first, then body	As it is
Ours	Marker-based referential	Bounding box	Floating menu, hand-menu system	Textual input, audio & video, picture, ontology matching, virtual objects	Target first, then body	Graph, Window, As it is, List Menu

on the depth camera of the MR device that the user could interact with in order to place annotations created by a remote user, which would interact and communicate through a tablet device

HandyNotes exploit some features that are present in your MR annotation systems. Nonetheless, we contribute with an more extensive investigation of the advantages and limitations of each technique. Previous annotation systems have a limited sample of content types that one can create and add to the annotation, and most of them do not provide clear information on how the VE is calibrated to support annotation of physical objects, or rely entirely on the existence of virtual objects. *HandyNotes* implements a simple, yet powerful, calibration technique through a marker-based referential, allowing (i) direct mapping to the VE coordinate system and (ii) sharing of annotations by anchoring them on the marker. Furthermore, we support a wider variety of types of content to enrich the annotations.

3 Design Rational of annotation task in MR

In order to understand the challenges related to the design of annotation systems using MR technology, we propose a rational design that is focused on annotation tasks (and the their corresponding sub-tasks). At the highest level of abstraction, an annotation is defined as a set of interconnected resources comprising an interrelated body and target [33]. Formally speaking, the target refers to the element we want to annotate, while the body's content often refers to the target [46]. Furthermore, annotations can contain metadata contextualizing the body's content and can assume diverse shapes (e.g., text, sketching, highlighting, etc.) and be attached to different artefacts (e.g., documents, images, data sets, etc.) [21]. Therefore, an annotation task consists of three basic sub-tasks: (i) target selection, (ii) content creation, and (iii) target-content association.

Nonetheless, the use of MR technology for annotating real world objects imposes the inclusion of special tasks such as the calibration of the virtual and the real environment, so that digital annotations can be traced to real world positions. Some tasks might require adaptations to tackle occlusions both in the real world and the virtual environment (VE). Moreover, MR technology supports multiple ways of creating annotations (e.g. using audio when writing is not possible), whose advantages and limitations might vary according to the context of use. Therefore, we revise the annotation task model using MR technology as follows:

1. **Calibration of VE**: this task is required to make the VE to match objects on the real world.
2. **Target creation**: this tasks allows to place interactive holograms in the VE that overlay real-world objects.
3. **Body creation**: this task creates the body content (semantic description featuring a text, an image, etc) of an annotation.
4. **Target-Body association**: this tasks connects the target and body, which otherwise would exist independently in the VE.
5. **Visualization of annotations**: this task is meant to help users to see and interact with all the annotations in a scene, even annotated objects are out of the users' field of view.

There are many possibilities to implement the annotation task in MR. In this section we discuss the different tasks necessary to achieve annotation in MR environments, the possible implementations of those tasks and the lessons we learned in the process. Table 2 summarizes the multiple design alternatives and how they match the following criteria: (i) **articulatory distance**, defined as the physical effort between the interaction and its meaning, (ii) **semantic distance**, is the distance between the user's intention and the meaning of expressions in the interface, and (iii) **feasibility**, refers to whether the proposed solution is doable through Microsoft Hololens 2. Design alternative are rated according to a three-point scale (0 - no support, 1 - limited support, and 2 - support) in terms of how much each approach supports those criteria.

Table 2: Summary of design rationale in terms of tasks and a classification of design alternatives according to a three-point assessment scale (0 to 2) regarding articulatory distance on input (AD-In) and output (AD-Out), semantic distance on input (SD-In) and output (SD-Out), and feasibility.

Task	Design Approach	AD-In	AD-Out	SD-In	SD-Out	Feasibility
Calibration of VE	Geo-localization by GPS	2	2	2	0	0
	Virtual Room	0	0	0	0	1
	Marker-based referential	0	2	1	2	2
Target Creation	3D object recognition	2	2	2	2	0
	3D assets library	1	0	1	1	2
	Mid-air drawing	0	2	0	0	1
Body Creation	Floating Menu	0	0	0	0	2
	Hand-Menu System	2	2	1	1	2
	Textual Input	0	0	2	2	2
	Audio and video	1	2	2	2	2
	Picture	2	2	2	2	2
	Ontology matching	1	2	2	1	2
Target-Body Association	Target first, then body	2	1	1	1	2
	Body first, then target	0	0	1	1	2
	Random	0	0	0	0	2
Visualization	Graph	2	0	1	1	2
	Window	1	2	1	1	2
	Annotated Environment	0	0	1	1	2
	List Menu	0	0	1	1	2

3.1 Calibration of VE

We explored the usage of three distinct approaches to support VE calibration: (i) geo-localization through GPS, (ii) spatial delimiters to define the annotation space, and (iii) a marker-based approach using QR codes as a referential.

Geo-localization by GPS Calibration the location of virtual objects through geographic coordinates (altitude, longitude) in the world reference frame may seems intuitive. Nonetheless, to the best of our knowledge, current MR devices do not have built-in GPS or cellular connectivity, which limits its ability to provide real-time geo-location data. External devices (GPS modules), or services (such as Google Geolocation API service with Magic Leap One or Windows Geolocator app with Hololens 2) might provide (or derive) the geo-location information through Wi-Fi connection. While the former adds supplemental hardware, making the device heavier and therefore less comfortable, the latter relies on a stable Wi-Fi connection, which would not allow offline annotation. The criteria of feasibility has low support as it would provide poor results at the cost of a more complex and limited application. **This approach has great support in terms of articulatory and semantic distances on input**, as there would be no effort from the user, since the system would automatically geo-locate the VE. However, **the approach does not support semantic distance on output**: either the system displays supplemental information to indicate that the calibration is done, which can be disturbing, or the user has no clear information on whether the calibration was a success.

Virtual room Inspired by the use of virtual walls in VR [9] we explored the usage of a virtual room delimited directly by the user to determine the annotation space. In this solution, users can define the boundaries of their workspace by placing pillars in each corner of the VE. A first pillar would define the height of the room, while the remaining would serve to position the corners on the X and Z axes. By using the measures of only one pillar, the goal was to provide a rapid control of the annotation space. Once the first pillar has been placed, the remaining ones can be quickly placed by following the perimeter of the room. To provide more control over the room, the user could also add and delete pillars at any time. To reduce the impact of user error, the system would guide the user in corner placement. While the solution is feasible, **the approach has no support in terms of articulatory and semantic distances**. The user faces a high learning curve before starting annotating the space such as creating virtual objects, accurately positioning them, and understanding how the collection of pillers calculate the delimited space. Furthermore, the user must either move around the room to place the pillars correctly or learn how to use ray casting to place pillars from a distance, which has a lower precision as the distance increases, and is susceptible to shaking of the hand. Multiple verifications and modifications might be necessary to ensure that the pillars are correctly placed and that the virtual room is correctly defined, which significantly increase the output semantic and articulatory distances.

Marker-based referential Marker-based approaches through QR codes provide a flexible and light solution: they do not require additional devices and the data can be directly processed by the MR device [1]. In this context, the physical location of the QR code serves as the center of the coordinate system, which would be used as a reference to locate each virtual object created by the user. **The approach is completely feasible** and recommended to be used with Microsoft Hololens [28]. A few limitations with this approach lie in the fact that objects are related to the QR code, which means that the system cannot detect when an object of the real world has been relocated, making the VE inconsistent with the real one. It also requires extra care to support continuous annotation of an environment through multiple sessions as the QR code must always be scanned in the same physical location to ensure the correct rendering of virtual objects. **The criterion of articulatory distance on input is not supported**, as the user is required to look for and scan the QR code before interacting with the environment. **The semantic distance on input has great support**, as little effort is necessary to understand that a QR code should be scanned and it can be rapidly learned. On output, **both semantic and articulatory distances are supported by the approach**, as a hologram overlays the QR code directly in front of the user's field of view, providing direct feedback on the calibration status.

3.2 Target creation

To create the target of an annotation, we explored three approaches: (i) 3D object recognition through AI algorithms that scan the real-world and directly overlay the objects with interactive holograms, (ii) a library of 3D assets from which the user can choose to represent the real-world object by placing it at the same location as the latter, and (iii) drawing holograms with mid-air gestures.

3D object recognition **This solution has great support in terms of articulatory and semantic distances**, as there is no effort required to create and place the holograms in the VE. However, 3D object recognition is still underdeveloped in current MR devices, making the approach unfeasible. While Microsoft Hololens 2 integrates spatial mapping algorithms, which supports real-world mapping through depth-sensing cameras allowing holograms to interact with real-world objects and surfaces, the current implementation of these technologies is far from optimal. For instance, Hololens 2's tracking performance is sensitive to the environment [27]. It can struggle in very bright or dark spaces. Reflective surfaces may yield inconsistent scans, and a stable Wi-Fi connection is crucial for accurate tracking. Without Wi-Fi info, recognition may be slower, and significant Wi-Fi signal changes can confuse the device's understanding of its location. Thus, although spatial recognition is possible, it is not enough to support 3D object reconstruction. Furthermore, irrespective of the specific device, it is important to recognize that spatial mapping data represents the most resource-intensive data source that applications can employ. Utilizing it demands increased application memory and may introduce potential overhead due to additional graphic and physics processing, particularly when used continuously. AR development platforms such as Vuforia could be used together with MR devices to support object recognition [15]. In this scenario, Vuforia could be used to recognize physical objects or images in the environment when viewed though the Hololens 2 camera, which triggers holographic overlays or interactions related to the recognized objects. Nonetheless, such an approach requires a prior preparation of the environment where every object to be recognized is assigned a specific image or QR code, limiting the note-taking process to rooms and objects that have been tagged accordingly.

3D assets library **Using a library of 3D assets has great support in terms of articulatory distance**, as the user only has to choose an object from the library and place it in the VE, which appears directly where the user placed it. Particularly, inspired by Wong et al. [47], we use bounding boxes as virtual objects to support interaction with the real world, as they allow for interaction and information display in a standard manner throughout objects. When annotating the real-world, we deal with different shapes and sizes of objects that should be visible at all times. Thus, the approach should support functionalities to resize and freely place the bounding box in the environment to match the position of the real object the user wants to annotate. The user should be able to create multiple bounding boxes, as well as to place them inside one another, e.g. to annotate a part of an object. The approach is completely feasible with Microsoft Hololens 2, as it facilitates the creation and manipulation of cube-shaped objects through the Mixed Reality Toolkit 2 to support ergonomic and smooth integration in the MR environment. **This approach provides limited support to the criteria of articulatory and semantic distances on input** as the user must select an object from the gallery, which may not correspond to the object they have in mind. For the same reason, **the criterion of semantic distance on output has also limited support**. However, the approach **has no support in terms of articulatory distance on output**, as much effort is required to resize and reposition the selected 3D asset on top of the physical object they want to annotate.

In a positive take, as the approach uses standard shapes, the learning curve is rather flat.

Mid-air drawing With a pinch gesture the user could draw a shape in the air, which would become an hologram representing the physical object underneath. This approach has been explored in previous works, where it has been appreciated by the users [12], but has been shown to have low accuracy due to the lack of physical support [39]. **The approach does not support articulatory and semantic input distances**, as the user must learn the drawing actions and prepare the gesture prior to drawing, which requires a great amount of effort to surround the real-world object, particularly when dealing with large and complex objects. **The articulatory distance on output has great support** as the object would be drawn directly in front of the user, requiring no effort to identify where the hologram has been drawn. However, **it does not support semantic distance on output** as the user would have to keep in mind the different holograms and what real-world objects they represent.

3.3 Body creation

The task of creating the annotation's body requires defining the content type (text, audio, video, pictures, tags) and the interaction that supports the creation of that content. Hereafter, we discuss two menu paradigms we investigated to support body creation, and five content types.

3.3.1 Body creation through menus

Floating Menu Floating menus are widely used in VR applications, where the user needs to make the 3D cursor intersect the appropriate menu choice. It has been shown to increase the possibility of making errors and it is often out of reach, which hinders selection by directly touching the menu items [16]. Indeed, **the approach does not support semantic and articulatory distances, both on input and output**. There is a difficulty to understand the relationship between a floating menu and the objects in the scene, due to its distance and visual disconnection to the holograms. When implemented on a fixed position, locating the menu requires multiple head movements, while when implemented in the user's field of view, it can be disturbing as it may hide the object-related holograms with which the user would want to interact. On output, **the articulatory distance is not supported** as upon command selection on the menu, the user must move around to locate the object that has been affected by it.

Hand-Menu System Due to the difficulties encountered using a floating menu, we decided to explore a hand-menu system [34] where the menu items are placed on the user's fingertips and, thus, move together with the user's hand. On input, **the approach supports articulatory distance on input**, as the menu is easily activated through a *palms up* action and the menu items can be selected through direct touch using the index finger of the other hand. **On output, the articulatory distance is supported** by bringing the menu next to the target hologram. **The support of semantic distance is limited**: there is a an initial effort necessary to understand the menu setup and how it relates to the objects in the scene.

3.3.2 Body content

Textual input Although textual input seems to be a fairly simple feature, it can be tricky to implement in a ergonomic manner within MR applications as we are often interacting with holograms that do not provide physical feedback. Hands-free interactions such as speech recognition are one of the most natural ways to input text in MR as users can speak directly into the microphone to input text. Guo et al. [20] showed that the Hololens 2 can correctly recognize about 80% of short voice commands (e.g. "select", "place", "remove") . However, human speech signals vary across speakers, speaking styles, content, and uncertain environmental noises which leads to speech recognition systems with low accuracy and accessibility [6]. The most common solution consists of using virtual keyboards, which are easily supported by current MR frameworks and allow users to select letters, numbers or symbols that are transposed to the textual input. Due to the familiarity of users with textual input, **the semantic distance is is greatly supported by the approach**. However, the approach does not support articulatory distance as the user must type the text in the air (letter by letter), which can be particularly tiring specially when dealing with long texts.

Audio and video The built-in cameras and microphones of MR devices facilitate content creation through audio and video recording. Furthermore, due to the familiarity of the approach to users, **it supports semantic distance on input and output**. While being easier than typing text into the air, recording audio and video requires a mild effort from the user. In particular, audio would require the user to speak out loud for a while and video would require head movements to record the real world, meaning that **it has a limited support in terms of articulatory input distance** necessary to perform the task. Nonetheless, **the approach supports articulatory distance on output**, as the resulting recording would be directly displayed to the user through the MR device camera.

Picture Similarly to video content, the built-in camera of MR devices support easy image capture. This approach helps to enrich the annotations with pictures of the real object. Due to the easiness of use and familiarity with the approach, **it supports both articulatory and semantic distance on input and output**.

Ontology matching With the advent of the Semantic Web, an interesting possibility is to enrich the annotations with named entities, which allows the classification and unique identification of entities such as person names, physical addresses, medical terms, etc. [18], that can later on be used to retrieve more information from RDF (Resource Description Framework) datasets on the Web. We explore the use of this approach by allowing the users to characterize the annotated objects through a set of pre-defined tags that represent the named entities. **Semantic distance on input is supported** as the use of tags for the purpose of classification is quite familiar to users, however on output the support is limited as users must agree on the meaning of the tag in association to the object being annotated. **The approach supports articulatory distance on output**, as the system includes the tag automatically in the annotation requiring no effort from the user. However, **it provides limited support to the articulatory distance on input** as it requires the user to choose and select a suitable tag within a list of tags.

3.4 Target-Body association

In an annotation system, the association between target and body is often performed in terms of selection order. Hereafter, we discuss three possibilities of selection order: (i) target first, then body, (ii) body first, then target, (iii) random selection order.

Target first, then body Since an annotation is often created with a target in mind, the natural approach is to select the target then create the body that will be automatically associated to that target. On input, **this solution has limited support to semantic distance** as the user must learn the selection order, and **it supports articulatory distance** as the user does not need to identify where the target and body are placed in the environment to select and associate them. **On output, the approach has limited support for both articulatory and semantic distances** as the user must understand and locate the association in the VE, which could be hidden by the real-world object or the target itself.

Body first, then target Conversely, users might prefer creating the annotation body before associating it to a target. For instance, let us assume a scenario where the user can import previously created digital annotations (e.g. from another compatible system). The

user would want to select all the annotations at once and then select the target to which they should be associated. **This approach does not support articulatory and semantic distances on input** as the user would have to search for the target hologram, which information would have to be extracted from the semantic meanings of the bodies. **On output, the approach provides limited support to articulatory and semantic distances**, as the user would have to understand what happened to the bodies (i.e. how they were linked to the target) and locate them in the environment.

Random Another solution would be to allow the user to create as many targets and bodies as they want in the environment without a particular relationship, then select two elements to associate them. **This solution does not support articulatory and semantic distance**. The user would have to identify which hologram is target and which is body, while analyzing the content of the latter to determine for which target it was intended. The annotation process would take longer and the outcome could be highly inaccurate.

3.5 Visualization of annotations

The system should support the visualization of annotations at the object and VE level. To visualize the annotations pertaining to a particular object, we explored two solutions: (i) a graph where holograms representing each annotation are linked to the target hologram and (ii) a window next to the target hologram regrouping all associated annotations. To visualize the set of annotations of the environment, we explore the usage of (i) a list menu, where the user can scroll and identify the different annotations in the VE, and (ii) an "as it is" approach, which presents the VE with the target and body holograms as they were created.

3.5.1 Visualizing an object's annotations

Graph The graph visualization has the target hologram in the center and the body holograms surrounding it connected by line segments. **This approach has limited support to semantic distance both on input and output** as it requires an initial effort to understand where the annotations are displayed and what the graph represents. As the number of annotations increase, more space is necessary to display them, which can rapidly clutter the VE. Thus, **the approach does not support articulatory distance on output**, as there is an effort to find the annotations and the user might need to look towards multiple locations to make sense of the visualization as a whole. Conversely, **the approach supports articulatory distance on input** as the links are automatically added to environment connecting the annotations and the corresponding bounding box.

Annotation Window Using a window that regroups the annotations next to the target reduces the space necessary to display all the information. Thus, **the approach supports articulatory distance both on input and output** as the annotations are automatically included on the window and everything is contained on the user's field of view. As for the graph, **the approach has limited support in terms of semantic distance** as one needs to understand where the annotations are displayed and what the window represents.

3.5.2 Visualizing the annotated environment

As it is The straightforward solution is to display the annotations as they were created, i.e. the bounding boxes at the target's positions together with the annotation window. As the holograms and annotations are already created in the MR environment, there are no issues in terms of feasibility. In terms of semantic distance, the approach requires an initial effort to understand that the holograms represent previous annotations, which limits its support. **It does not support articulatory distance neither on input nor on output** as the user must locate and move towards the annotations.

List Menu Using a list menu allows the user to have all the annotations of the VE in a single place. Adding 2D elements such as lists in an MR environment is a fairly simple task, which facilitates development. On input, since users are familiar with lists, this approach supports semantic distance. However, it does not support articulatory distance on input as the user must scroll down the list to identify the annotation they are looking for. Furthermore, previous work have shown that scrolling within a MR menu can be frustrating and cumbersome [8, 38]. On output, it does not support articulatory distance as, after selecting an annotation in the list, the user must locate the annotated object in the room, which can be hidden within the user / another object, hindering the location task or it could be far away, requiring physical displacement. For this same reason, the approach provides limited support to semantic distance. Using mechanisms such as highlighting the selected annotation in the room could improve the semantic distance necessary to understanding that the selection worked and that the annotation is placed elsewhere.

4 HandyNotes: System Description

Our system was built upon Microsoft's Hololens 2 headset, which includes the latest Kinect sensor, a custom AI chip to improve its performance, a wider FOV (5̃3 degrees diagonal, 43 horizontal, 29 vertical), inside-out tracking (no external sensors required), holographic lens integration, and a fairly comfortable design. The development employed (i) the Unity version 2020.3.2f1, recommended by Microsoft for development on HoloLens 2, and (ii) the Mixed Reality Toolkit (MRTK2).

4.1 Calibration of VE

Upon startup, the application requires the user to scan a QR Code to calibrate the VE to the real environment. The system is able to load the VE with the associated annotations, , otherwise, the systems creates a new VE that will be associated to that QR code. The user starts annotating the environment by creating new annotations or modifying the existing ones.

4.2 Hand-Menu System

A hand-menu system is anchored on the user's non-dominant hand and can be revealed by a *palms up* gesture (Figure 3). We represent the menu items through spherical buttons associated to different commands. Each button contains an icon relative to the command their activate, e.g. create a new bounding box, add a textual annotation, etc. Each item provides feedback through color change, sounds and push-button animation to inform the user when an item has been selected. These items are placed at the fingertips and on certain palm joints to space the buttons apart. In addition, the buttons are oriented towards the user, so that they can always clearly identify the different icons and descriptions of each item regardless of the hand's inclination.

Our Hand-Menu System comprises three states:

- **Default:** As the name suggests, this state is enabled by default on start up , with no hologram selected in the VE. Here, the menu items allow the user to create virtual holograms, which can be enabled and disabled at will (Figure 3a). The **Bounding Box** feature allows to create of a cube-shaped bounding box, which is used to define the annotation target by placing it on the top of the physical objects. The **Floating tag** item allows the user to annotate sub-parts of a target an anchor ball linked to a text, by placing it withing a bounding box, e.g. to annotate a button in a remote control (Figure 2a). The **Virtual Object** item allows the user to create "semantic" virtual objects such as chairs, tables, cars, etc. that can better express the meaning of the annotated object in the real world.

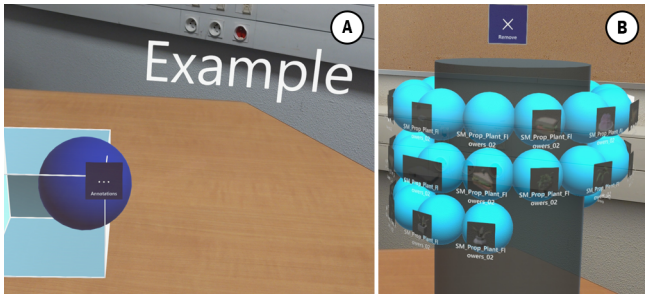

Figure 2: (a) A floating tag linked to a bounding box. (b) Cylinder of virtual objects.

- **Annotation:** This state is enabled when the user selects a bounding box in the VE, it contains items to support body creation. The user can create multiple content types: text, audio, video, picture, and ontology matching (Figure 3b). Further, the item **Link** allows the user to create line segment that connects two annotation targets, e.g. to link a remote control to a TV to indicate that the remote belongs to that TV.

- **Settings:** Finally, this state (Figure 3c) provides settings options such as: **save** the state of the currently annotated environment, which VE is then associated to the QR code used during the calibration phase, **load** a previously annotated environment by using a previous QR code, and **visualize** the annotated environment. The latter allows users to explore the existing annotations without risking unwanted modifications due to parasite actions, such as accidental gestures.

4.3 Interaction through gestures

The interaction language is based on air gestures, which are defined by the position of hand joints. We use the MRTK2 toolkit to track the hand joints, which tracks the position of each joint and assign a state, ranging from 0 (finger closed) to 1 (finger extended). We implemented the following air gestures :

- the *index gesture* allows the user to interact with virtual elements by directly touching them.

- the *pinch gesture* is used for geometric manipulation, such as translating, rotating, or resizing virtual holograms. It can be performed by both hands; for example, doing a translation with one hand, and simultaneously resizing an object with the other.

- the *palm up gesture* supports distinct functions according to the hand used to perform the gesture and the interaction context. The non-dominant hand brings up the hand-menu system, which function changes according to the interaction context: a palm up gesture when there is a bounding box selected reveal the body creation menu, otherwise it brings up the target creation menu. The dominant hand creates a bounding box when nothing is selected. When an object is selected, the gesture triggers the display of a menu showcasing all annotations directly in front of the user.

- the *Hand ray* is used to select holograms which are out of physical reach or occluded by a physical object, e.g. a remote control behind a box. Hand ray is a ray cast from the center of each hand to target a distant object. It is activated by a partially closed hand, like holding a ball in the hand. When the hand is fully closed, the ray allows you to (i) select a bounding box or menu interface, or (ii) click on a remote button.

4.4 Target creation

Upon identifying an object of interest in the real world, the user can invoke a bounding box through a *Palm Up gesture* on the dominant hand, which displays a cube-shaped object on the top of the palm. The bounding box has four states: (i) geometric manipulation, (ii) annotation-ready, (iii) inner-annotation, and (iv) inactive. Upon creation, the bounding box is on *geometric manipulation* state, which is indicated by a green color. In this state, the user can resize it by grabbing a corner through a *pinch* gesture or *hand ray* and drag it to reach the appropriate size (e.g. the size of the physical object). Similarly, the user can grab an edge of the box and drag it to translate or rotate it, to match the position and rotation angle of the physical object. Once the bounding box has been correctly positioned and resized, the user should confirm the transformations by touching the bounding box with an *index* gesture, or by pressing the confirmation button on the hand-menu system. Upon confirmation, the system takes a picture of the physical object to keep a reference of it throughout the annotation process. Then, automatically, the bounding box enters on an *annotation-ready* state, which is indicated in blue. In this state, the user can use the options in the *Annotation* state of the hand-menu system to create the annotation body.

The user can annotate objects inside bounding boxes. For example, let us assume that Alice has annotated a shoe box, for which she created a bounding box A, and decides to annotate each of the shoes inside it. To achieve this, from either the *annotation-ready* or *inactive* states, she inserts her hand into the bounding box A, which switches to the *inner-annotation* state, indicated in yellow. In this state, collisions disabled to allow the user to create new annotation targets and bodies through the process described above, while manipulating the new bounding boxes within the bounding box A. Collision deactivation is essential to avoid any undesirable behavior when manipulating objects inside it. This enables the user to perform geometric manipulations of the bounding boxes inside without any interference from the external bounding box. Once all hands are removed from the bounding box A, it returns to its starting state, i.e. *annotation-ready* or *inactive*, and collisions are reactivated.

Whenever a bounding box is not being used (i.e. when the user creates another bounding box), it enters the *inactive* state. The user can resize and move the bounding box at all times. From either the *annotation-ready* or *inactive* states, the user can perform a *pinch* gesture or *hand ray* on the box for one second to activate the *geometric manipulation* state. Similarly, at all times, the user can invoke a new cube through a *Palm Up* gesture, which automatically changes the state of all bounding boxes to *inactive*.

4.5 Body creation

Once a bounding box is selected, the user can perform the gesture *palm up* with the non-dominant hand to show the menu allowing the creation of the following body types:

- **Text:** The user can click on the text icon on the menu, which triggers a a virtual keyboard and an input field directly on the user's field of view. The user types the desired text and confirm it by clicking on a button on the top of the input field. The virtual keyboard automatically disappear after confirmation.

- **Audio:** The users press the *audio* item in the hand-menu system, which activates the HMD's microphone and initiates the recording process. While recording is in progress, the system replaces the icon on the audio button with a stop icon, the user can press it again to stop the recording. For storage restrictions. Audios cannot exceed 30 seconds. The audio is automatically added to the annotation window once it is complete, and the system replays it, to inform the user that everything went well.

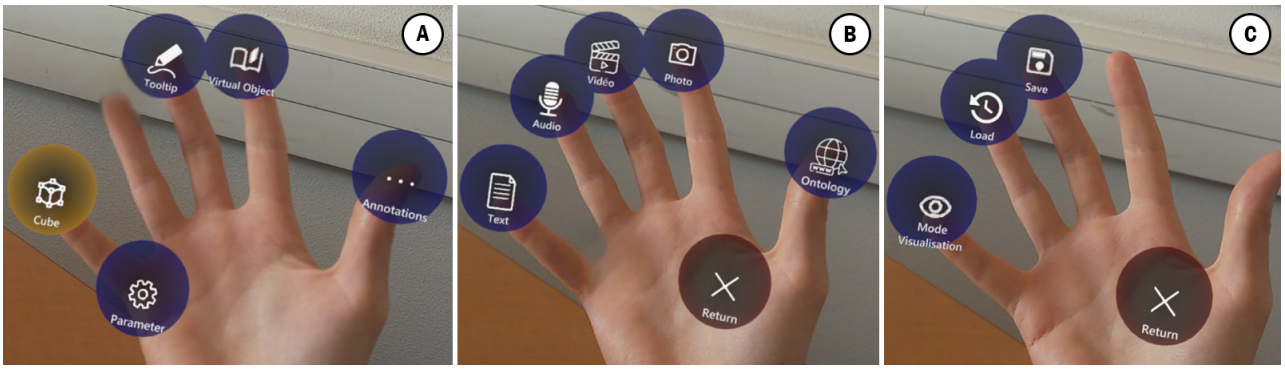

Figure 3: Overview of the Hand Menu System. (a) **Default** state, where the user can create virtual holograms such as bounding boxes, directions, floating tags. (b) **Annotation** state, which contains items allowing body creation through links, text, audio, video, pictures, and ontology matching. (c) **Settings** state, where user can save the current annotated VE, load previously annotated VEs and visualize the annotations.

- **Video:** Video annotation is similar to the audio, the user clicks on the video item on the menu to start the recording and clicks on it again to stop the recording. We use the HMD's built-in camera and the user must move around and look towards what they want to record. Video cannot exceed 30 seconds.

- **Picture:** Using the HMD's built-in camera, the system can create a picture of whatever is the user's field of vision. After locating the target, the users must click on the photo item on the hand-menu system to take the picture. Then, the system takes the control of the camera, wait 2 seconds to prevent the application from recording an empty photo, and take the photo, which is automatically added to the annotation window.

- **Ontology matching:** To associate pre-defined tags to the annotated object, the user can use the *Ontology matching*, which is composed of predefined tags, disposed around an interactive cylinder. To activate it, the user clicks on the ontology button in the hand-menu system, which triggers a cylinder positioned next to the active bounding box. By scrolling the ontology, such as on a carousel, the user can rotate and translate the cylinder to identify suitable tags for describing the object. Users can directly select a tag of interest with an *index gesture*, automatically adding them to the annotation. Deleting tags from the annotation is easily done through the annotation window.

- **Virtual object:** From a gallery of objects, the user can select one to enrich the VE and/or to indicate a missing object in the virtual world. Creating virtual objects is similar to creating bounding boxes, i.e. using the Virtual Object item in the Default menu. Then, a *palm up gesture* will trigger a virtual cylinder containing a set of possible virtual objects (Figure 2b). Similar to the ontology matching widget, the user can move and rotate it, and then select the virtual object they want to add to the VE. After selection, the system replaces the cylinder with the chosen virtual object, which can then be edited using the same manipulations and gestures as the bounding box.

- **Floating tags:** To create a floating tag annotation, the user switches the hand-system menu to its *Default* state and selects the floating tag item. It consists of a textual information anchored to a ball. A floating tag can be placed within a bounding box, which binds the tag to the box, or anywhere in the VE, which will result on an independent tag, i.e. not associated to any object. For bound tags, the anchor moves with the bounding box whenever it is re-positioned in the VE. The user can move the tag by using the *Pinch gesture* on the text and do the same with the anchor by grabbing the ball.

4.6 Target-body association

We implemented a *target first, then body* approach to support target-body association. Thus, when a bounding box is on the *annotation-ready* state, the hand-menu system automatically shows the buttons allowing the user to create the body, which is automatically associated to the target. The annotation body is displayed directly on the annotation window next to the corresponding bounding box. To reduce mismatches between target and body, the user can only annotate one bounding box at a time.

4.7 Visualization of annotations

In order to manage and display the annotations of each bounding box, the system displays an annotation window next to it (Figure 1c). It rotate according to the user's gaze, i.e. regardless from which angle the user looks at the bounding box, the window is in front of it, preventing occlusion with the bounding box. The window is created when the bounding box enters for the first time on the *annotation-ready* state. It presents all the annotation content associated to that target, and can be used to delete the annotation target through a delete button on the top. The window is destroyed at the same time as the bounding box. The user can preview the content by clicking on the corresponding item on the panel at the right side of the window. During preview, the user can also delete the content if they wish to do so. When the bounding box is not on *annotation-ready* state, the window is minimized into a three-dot button floating next to the bounding box, indicating that the object has associated annotations.

4.8 Log file and close up

The user can save the annotated VE at any time using the save button from the *settings* state of the hand-menu system. The system also saves the VE data on exit, to prevent it from crashing and losing information in case the user forgets to save. The annotated VE data is saved in a `.dat` file named after the QR Code identifier. Each QR code contains a unique identifier which allows us to associate one file per QR Code and, thus, to retrieve previously created annotations in that environment. The file follows a JSON structure, which contains the objects' identifier, name, position according to the QR code position, rotation, scale and the path to the associated annotations with the time when they were created.

5 Use Case Scenarios

Use Case Scenario 1 In this scenario HandyNotes is used to support a maintenance task by placing annotations with instructions for making repairs. Figure 1 illustrates how the user can annotate a piece of broken equipment such as a video projector and its accompanying remote control. The user creates a bounding box with

the *Palm Up* gesture and then uses the *pinch* gesture to resize and rotate the bounding box to match the physical object. As the video projector is placed on the ceiling, the user must to use the *hand ray* to place the bounding box on the projector's physical location (Figure 1a). The user then confirms the bounding box's position with the hand-menu system. The system takes a picture of the video projector, which appears in the annotation window (Figure 1b). As the bounding box enters *annotation-ready* state, the user can add a textual and an audio description of the object. In a related note, these same steps are followed when the user wants to annotate a hidden object, which is simply treated as a distant object. The user then repeats the annotation steps for the remote control. Once all objects of interest are annotated, the user uses the *settings* state in the hand-menu system to save the annotated VE with the QR code. This annotated VE can now be opened by other users to understand how to use the projector and remote together.

Use Case Scenario 2 In this scenario, our system serves as a navigational guide for users, offering directional information through features like voice guides or images. Let's assume that our main user is organising a reception on a large campus. The organizers can use the tool to strategically place virtual annotations to direct participants towards meeting rooms, services, and facilities. By scanning a QR code located at the building's entrance, attendees have access to the annotations. Using Hololens 2, the user reveals the hand-menu system and switches to *settings*, where they can activate the *Visualization mode*. Notice that annotations created using our tool can also be exported (as text) and become available for users that do not possess a Hololens 2.

6 User trial

We ran a user trial to gather feedback from potential users of our system. We invited four students to use HandyNotes to perform tasks described in the Case Scenario 1. The participants were provided with informed consent on the experiment's objectives, procedures, and potential risks, which they signed to take part in the study. Since the system requires a long use of hand and arm movements, we asked them whether they felt any pain or had any medical issues regarding their lower arm, for which they all reported no pain nor medical conditions. Three participants were right-handed, and one was left-handed. They self-reported normal or corrected vision. None of the participants had prior experience with MR applications before joining the experiment. Prior to using HandyNotes, we asked users to play with Microsoft's Astuce application to practice MR interactions. After performing the Case Scenario 1, we asked participants to share their impressions of the system and complete a questionnaire pertaining to the cognitive and physical effort necessary to use the system. Overall, the annotation task took about 20 minutes.

All participants successfully completed the annotation task. Overall, they felt comfortable using the system. When performing geometric manipulation on the bounding boxes, users found it easier to use direct touch through *index gesture* over *hand ray*, as it would yield a more accurate result according to their manipulation intentions. P1 found the ergonomic design difficult to understand. P1 and P4 both suggested adding a tutorial to assist users in mastering the interactions and remembering the hand gestures. P1 and P4 also indicated fatigue in the lower arm after the experiment due to the repeated and prolonged use of the hand-menu system. They suggested to simplify this menu.

The Hololens 2 allow users to quit an active application and return to the Home screen by tapping on their wrist (any arm), which can cause unwanted actions. For instance, P2 and P4 mentioned that, at times, while using the hand-menu system they would touch their wrists by accident, which would make them quit the application and return to the Hololens Home screen. This issue has been reported as disturbing and should be inspected in the future. Despite this, users generally found the hand-menu system to be user-friendly and enjoyable to use. P1 and P4 mentioned that they had difficulty perceiving the depth of different objects, which hindered the process of placing the bounding boxes correctly on top of physical objects. P3 also mentioned that when creating a textual annotation, the text input was too high with respect to their field of view.

7 Discussion, Conclusions and Future Work

This paper presents the system HandyNotes which allows to create semantic representations of contextually aware real-world objects using MR technology. We explored multiple design alternatives to support the annotation task through MR devices. We learned that, while GPS and 3D object recognition are powerful technologies to support easy understand and interaction with the MR environment, their integration in current MR devices is far from optimal, which led us to explore options such as markers and bounding boxes. Although these approaches might increase articulatory distance, they reduces semantic distance as users can rapidly get familiar with such tools, while being fully supported by MR devices. The positive results from the first user trial confirm that.

Due to the lack of space, we could only illustrate two use cases scenarios: for maintenance tasks and user navigation tasks. And yet, the scenarios and user trial demonstrate feasibility of our system. Nonetheless, further studies are necessary, in particular, we want to investigate how the menu-hand system contributes to arm fatigue and how we can make it more ergonomic. We also want to explore in the future, the use of eye-tracking to improve interactions such as button selection on the hand-menu system. We raise the hypothesis that [42], gaze could be less physically demanding and is therefore an interesting alternative if combined with voice commands or gestures, avoiding the Midas problem.

Another research path would lead to investigate the usage of voice commands to improve the *inner-annotation* state of our bounding box. Inner-annotation is rather straightforward in a near situation, but it can be tricky when dealing with far away objects, where all manipulations use hand ray. The hand ray targets the surface of the closest object, i.e. the outer bounding box. For instance, we could combine hand gestures and voice commands to target and disable the colliders, respectively, thus easing the interaction.

Consistent with previous work [34], we noticed that the Hololens 2 recognition system had troubles for distinguishing between the menu and the selection hand when they overlapped (i.e. the user uses the index finger to select a menu item). We noticed interaction issues such as buttons that moved to the other hand, and difficulties to select the item. We would like to investigate the impact of using semantic virtual objects instead of a cube-shaped bounding box. Specifically, we hypothesize that it could improve usability and reduce the need for multiple body content types for a single target. This virtual object would work similarly to the bounding box, which the user could resize, select, and attach annotations.

We suggest that the chronology of object annotations and the analysis of user interactions might provide further insights. Using logs, it would be also possible to visualize past positions and annotations . To do this, we would set up a path enabling you to retrace the position of the object and at different intersection points the historical information about annotation.

Ultimately, we aim to expand the annotation approach to different immersive situations, including VR. In this scenario, users have the capability to annotate virtual objects within a virtual world, offering valuable assistance in analytical tasks for data retention.

Acknowledgments

This was was partially supported by the institute IMREDD, the the Graduate School and Research DS4H, and Polytech Nice. We also would like to thank the students M. Swery, E. Radu, F. Météreau, M. Bouteiller, V. Losciale, and R. Karaki who contributed to this project.

REFERENCES

[1] M. Aleksy, M. Troost, F. Scheinhardt, and G. T. Zank. Utilizing HoloLens to Support Industrial Service Processes. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 143–148, May 2018. ISSN: 2332-5658. doi: 10.1109/AINA.2018.00033

[2] J. Amores, X. Benavides, and P. Maes. Showme: A remote collaboration system that supports immersive gestural communication. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, p. 1343–1348. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2702613.2732927

[3] J. Amores and J. Lanier. Holoart: Painting with holograms in mixed reality. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, p. 421–424. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3027063.3050435

[4] T. Azai, M. Otsuki, F. Shibata, and A. Kimura. Open palm menu: A virtual menu placed in front of the palm. In *Proceedings of the 9th Augmented Human International Conference*, AH '18. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3174910.3174929

[5] Y. Bao, J. Wang, Z. Wang, and F. Lu. Exploring 3d interaction with gaze guidance in augmented reality. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 22–32, March 2023. doi: 10.1109/VR55154.2023.00018

[6] S. Basak, H. Agrawal, S. Jena, S. Gite, M. Bachute, B. Pradhan, and M. Assiri. Challenges and limitations in speech recognition technology: A critical review of speech signal processing algorithms, tools and systems. *CMES-Computer Modeling in Engineering & Sciences*, 135(2), 2023.

[7] M. Bauer, G. Kortuem, and Z. Segall. "where are you pointing at?" a study of remote collaboration in a wearable videoconference system. In *Digest of Papers. Third International Symposium on Wearable Computers*, pp. 151–158, Oct 1999. doi: 10.1109/ISWC.1999.806696

[8] J. D. Benedict, J. D. Guliuzo, and B. S. Chaparro. The intuitiveness of gesture control with a mixed reality device. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63(1):1435–1439, 2019. doi: 10.1177/1071181319631403

[9] M. Boldt, B. Liu, T. Nguyen, A. Panova, R. Singh, A. Steenbergen, R. Malaka, J. Smeddinck, M. Bonfert, I. Lehne, M. Cahnbley, K. Korschinq, L. Bikas, S. Finke, M. Hanci, and V. Kraft. You Shall Not Pass: Non-Intrusive Feedback for Virtual Walls in VR Environments with Room-Scale Mapping. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 143–150. IEEE, Reutlingen, Mar. 2018. doi: 10.1109/VR.2018.8446177

[10] Z. Borhani. [dc] annotation in asynchronous collaborative immersive analytic environments using augmented reality. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 950–951, March 2022. doi: 10.1109/VRW55335.2022.00326

[11] W. Büschel, A. Lehmann, and R. Dachselt. Miria: A mixed reality toolkit for the in-situ visualization and analysis of spatio-temporal interaction data. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445651

[12] Y. S. Chang, B. Nuernberger, B. Luan, and T. Höllerer. Evaluating gesture-based augmented reality annotation. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 182–185, March 2017. doi: 10.1109/3DUI.2017.7893337

[13] G. Chantziaras, A. Triantafyllidis, A. Papaprodromou, I. Chatzikonstantinou, D. Giakoumis, A. Tsakiris, K. Votis, and D. Tzovaras. An augmented reality-based remote collaboration platform for worker assistance. In A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, eds., *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 404–416. Springer International Publishing, Cham, 2021.

[14] S. M. Cochran, C. A. Aiken, C. K. Rhea, and L. D. Raisbeck. Effects of an external focus of attention and target occlusion on performance in virtual reality. *Human Movement Science*, 76:102753, 2021. doi: 10.1016/j.humov.2021.102753

[15] J. Cýrus, D. Krčmařík, M. Petrů, and J. Kočí. Cooperation of Virtual Reality and Real Objects with HoloLens. In K. Arai and S. Kapoor, eds., *Advances in Computer Vision*, Advances in Intelligent Systems and Computing, pp. 94–106. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-17798-0_10

[16] R. Dachselt and A. Hübner. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics*, 31(1):53–65, Jan. 2007. doi: 10.1016/j.cag.2006.09.006

[17] A. Elor, S. Whittaker, S. Kurniawan, and S. Michael. Biolumin: An immersive mixed reality experience for interactive microscopic visualization and biomedical research annotation. *ACM Trans. Comput. Healthcare*, 3(4), nov 2022. doi: 10.1145/3548777

[18] J. Filali, H. B. Zghal, and J. Martinet. Ontology-based image classification and annotation. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(11):2040002, 2020. doi: 10.1142/S0218001420400029

[19] I. García-Pereira, C. Portalés, J. Gimeno, and S. Casas. A collaborative augmented reality annotation tool for the inspection of prefabricated buildings. *Multimedia Tools and Applications*, 79(9):6483–6501, Mar. 2020. doi: 10.1007/s11042-019-08419-x

[20] H.-J. Guo and B. Prabhakaran. HoloLens 2 Technical Evaluation as Mixed Reality Guide, July 2022. arXiv:2207.09554 [cs]. doi: 10.48550/arXiv.2207.09554

[21] J.-L. Hak, M. Winckler, and D. Navarre. PANDA: prototyping using annotation and decision analysis. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '16, pp. 171–176. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2933242.2935873

[22] F. A. Hansen. Ubiquitous annotation systems: Technologies and challenges. In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, HYPERTEXT '06, p. 121–132. Association for Computing Machinery, New York, NY, USA, 2006. doi: 10.1145/1149941.1149967

[23] J. Herbert and T. Herrmann. An ar-method for documenting lego serious play models. In *Proceedings of the 7th ACM International Symposium on Pervasive Displays*, PerDis '18. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3205873.3210703

[24] R. Horst, R. Naraghi-Taghi-Off, L. Rau, and R. Dörner. Back to reality: transition techniques from short HMD-based virtual experiences to the physical world. doi: 10.1007/s11042-021-11317-w

[25] H. J. Kang, J.-h. Shin, and K. Ponto. A comparative analysis of 3d user interaction: How to move virtual objects in mixed reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 275–284, March 2020. doi: 10.1109/VR46266.2020.00047

[26] Y. Lu, X. Wang, J. Gong, L. Zhou, S. Ge, et al. Classification, application, challenge, and future of midair gestures in augmented reality. *Journal of Sensors*, 2022, 2022. doi: 10.1155/2022/3208047

[27] Microsoft. Hololens environment considerations. https://learn.microsoft.com/en-us/hololens/hololens-environment-considerations, 2022. Accessed on September 28, 2023.

[28] Microsoft. Coordinate systems in directx. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/coordinate-systems-in-directx, 2023. Accessed on January 17, 2024.

[29] P. MILGRAM and F. KISHINO. A Taxonomy of Mixed Reality Visual Displays, 1994.

[30] O. Oda, C. Elvezio, M. Sukan, S. Feiner, and B. Tversky. Virtual replicas for remote assistance in virtual and augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, p. 405–415. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807497

[31] K. Pfeuffer, J. Obernolte, F. Dietz, V. Mäkelä, L. Sidenmark, P. Manakhov, M. Pakanen, and F. Alt. Palmgazer: Unimanual eye-hand menus in augmented reality, 2023. doi: 10.48550/arXiv.2306.12402

[32] S. Rokhsaritalemi, A. Sadeghi-Niaraki, and S.-M. Choi. A Review on

Mixed Reality: Current Trends, Challenges and Prospects. *Applied Sciences*, 10(2):636, 2020. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute. doi: 10.3390/app10020636

[33] R. Sanderson, P. Ciccarese, and H. Van De Sompel. Designing the W3C open annotation data model. In *Proceedings of the 5th Annual ACM Web Science Conference*, pp. 366–375. ACM, Paris France, 2013. doi: 10.1145/2464464.2464474

[34] H. Sasaki, T. Kuroda, P. Antoniac, Y. Manabe, and K. Chihara. Hand-menu system: a deviceless virtual input interface for wearable computers. *Journal of Control Engineering and Applied Informatics*, 8(2):44–53, 2006. doi: 10.18974/tvrsj.7.3
_393

[35] A. Schäfer, G. Reis, and D. Stricker. Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications. *Applied Sciences*, 12(4), 2022. doi: 10.3390/app12041888

[36] A. Schäfer, G. Reis, and D. Stricker. Comparing controller with the hand gestures pinch and grab for picking up and placing virtual objects. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 738–739, 2022. doi: 10.1109/VRW55335.2022.00220

[37] A. Schäfer, G. Reis, and D. Stricker. The gesture authoring space: Authoring customised hand gestures for grasping virtual objects in immersive virtual environments. In *Mensch und Computer 2022*. ACM, sep 2022. doi: 10.1145/3543758.3543766

[38] W. Sinlapanuntakul, J. Korentsides, A. M. Collard, K. S. Skilton, and B. S. Chaparro. Touching holograms: A preliminary evaluation of mixed reality gestures. 2022.

[39] T. Teo, G. A. Lee, M. Billinghurst, and M. Adcock. Hand gestures and visual annotation in live 360 panorama-based mixed reality remote collaboration. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, OzCHI '18, p. 406–410. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3292147.3292200

[40] M. Tikat, A. Menin, M. Buffa, and M. Winckler. Engineering Annotations to Support Analytical Provenance in Visual Exploration Processes. In *ICWE 2022 - 22nd International Conference of Web Engineering*, vol. LNCS - 13362 of *Web Engineering*, pp. 1–16. Bari, Italy, 2022. doi: 10.1007/978-3-031-09917-5
_14

[41] C. Vazquez, N. Tan, and S. Sadalgi. Home studio: A mixed reality staging tool for interior design. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411763.3451711

[42] P. Wang, X. Bai, M. Billinghurst, S. Zhang, X. Zhang, S. Wang, W. He, Y. Yan, and H. Ji. Ar/mr remote collaboration on physical tasks: A review. *Robotics and Computer-Integrated Manufacturing*, 72:102071, 2021. doi: 10.1016/j.rcim.2020.102071

[43] Z. Wang, H. Wang, H. Yu, and F. Lu. Interaction with gaze, gesture, and speech in a flexibly configurable augmented reality system. *IEEE Transactions on Human-Machine Systems*, 51(5):524–534, Oct 2021. doi: 10.1109/THMS.2021.3097973

[44] N. Weibel, D. Gasques, J. Johnson, T. Sharkey, Z. R. Xu, X. Zhang, E. Zavala, M. Yip, and K. Davis. Artemis: Mixed-reality environment for immersive surgical telementoring. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, p. 1–4. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3334480.3383169

[45] A. S. Williams and F. R. Ortega. Understanding gesture and speech multimodal interactions for manipulation tasks in augmented reality using unconstrained elicitation. *Proc. ACM Hum.-Comput. Interact.*, 4(ISS), nov 2020. doi: 10.1145/3427330

[46] M. Winckler, P. A. Palanque, J. Hak, E. Barboni, O. Nicolas, and L. Goncalves. Engineering annotations: A generic framework for gluing design artefacts of interactive systems. *Proc. ACM Hum. Comput. Interact.*, 6(EICS):174:1–174:36, 2022. doi: 10.1145/3535063

[47] Y.-S. Wong, H.-K. Chu, and N. J. Mitra. Smartannotator an interactive tool for annotating indoor rgbd images. In *Computer Graphics Forum*, vol. 34, pp. 447–457. Wiley Online Library, 2015. doi: 10.1111/cgf.12574