



GraV: Grasp Volume Data for the Design of One-Handed XR Interfaces

Alejandro Aponte*

aponte@ucsb.edu

University of California
Santa Barbara, CA, USA

Yunhao Luo

yunhaoluo@ucsb.edu

University of California
Santa Barbara, CA, USA

Arthur Caetano*

caetano@ucsb.edu

University of California
Santa Barbara, CA, USA

Misha Sra

sra@ucsb.edu

University of California
Santa Barbara, CA, USA

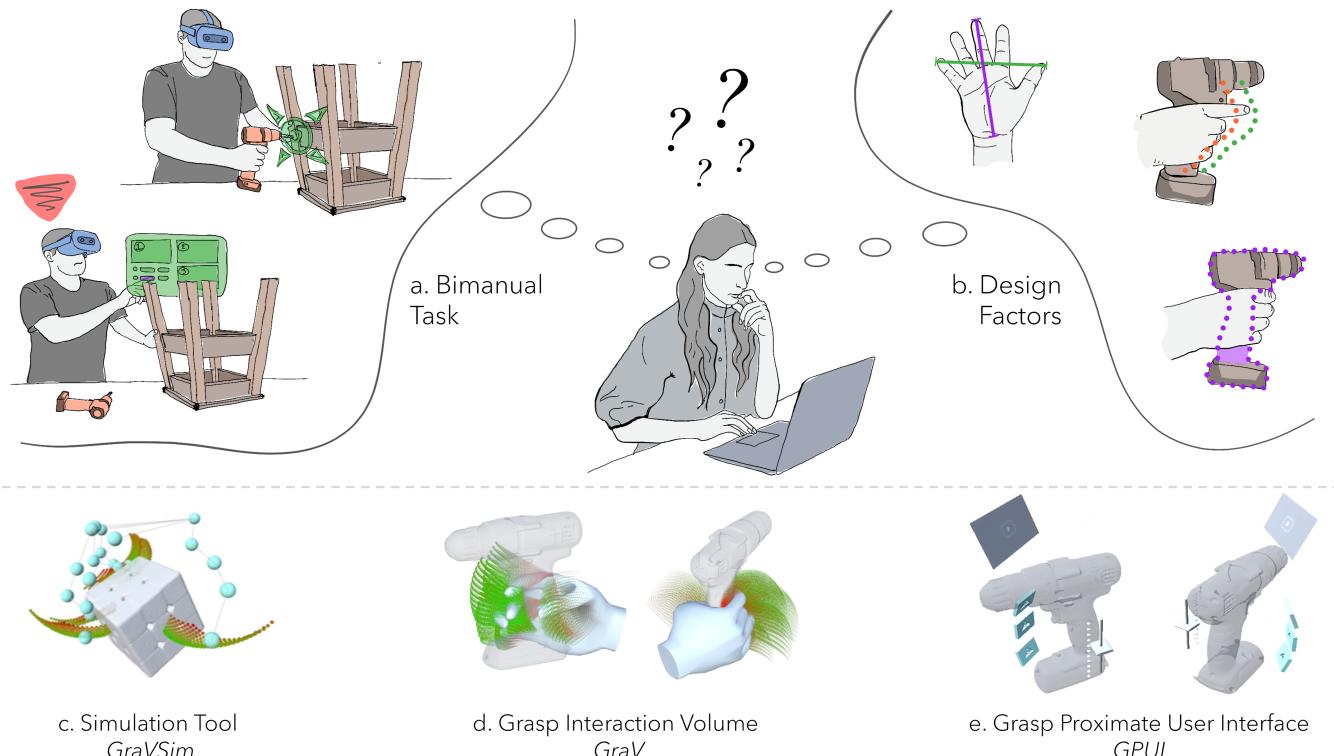
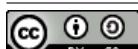


Figure 1: (a) When performing a bimanual task, users must place down the manipulated object to interact with mid-air interfaces, causing inconvenience and disrupting task flow. (b) Designers approaching this problem face a complex combination of parameters including hand size, motion range, and object shape. (c) GraVSim facilitates this process by simulating hand-object interactions to create GraVs (d), representing reach and motion cost. (e) Our tools simplify the design process of GUIs, interfaces that offer finger-accessible interactions while maintaining object grasp.

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution-Share Alike International 4.0 License.

DIS '24, July 01–05, 2024, IT University of Copenhagen, Denmark

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0583-0/24/07

<https://doi.org/10.1145/3643834.3661567>

ABSTRACT

Everyday objects, like remote controls or electric toothbrushes, are crafted with hand-accessible interfaces. Expanding on this design principle, extended reality (XR) interfaces for physical tasks could facilitate interaction without necessitating the release of grasped tools, ensuring seamless workflow integration. While established data, such as hand anthropometric measurements, guide the design of handheld objects, XR currently lacks comparable data, regarding reachability, for single-hand interfaces while grasping objects. To

address this, we identify critical design factors and a design space representing grasp-proximate interfaces and introduce a simulation tool for generating reachability and displacement cost data for designing these interfaces. Additionally, using the simulation tool, we generate a dataset based on grasp taxonomy and common household objects. Finally, we share insights from a design workshop that emphasizes the significance of reachability and motion cost data, empowering XR creators to develop bespoke interfaces tailored specifically to grasping hands.

CCS CONCEPTS

- **Human-centered computing → User interface design; Mixed / augmented reality.**

KEYWORDS

spatial user interface, dataset, extended reality, grasp-proximate interfaces

ACM Reference Format:

Alejandro Aponte, Arthur Caetano, Yunhao Luo, and Misha Sra. 2024. GraV: Grasp Volume Data for the Design of One-Handed XR Interfaces. In *Designing Interactive Systems Conference (DIS '24), July 01–05, 2024, IT University of Copenhagen, Denmark*. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3643834.3661567>

1 INTRODUCTION

Physical tasks like assembly, repair, or surgery often require the use of both hands. One hand holds a tool such as a screwdriver or a scalpel, while the other hand manipulates the object being worked on. To assist in these types of tasks, many devices feature hand-accessible interfaces while the user is grasping the object. These hand-accessible interfaces facilitate interaction with the system while enabling a more continuous workflow without releasing the device. XR interfaces have the potential to apply this strategy beyond the current form factor and intended use of the grasped object, by overlaying virtual UI elements surrounding the hand, irrespective of the grasped object's original design.

There has been a growing interest in the design of extended reality (XR) applications for physical task assistance [20, 37, 52]. Nonetheless, current XR interfaces are not well suited for tasks that occupy the user's hands. Researchers have proposed hands-free interaction modalities such as gaze and speech which have been explored successfully in XR (see [35, 39] for recent surveys). However, these approaches still have some limitations. Prior work shows that gaze-based interfaces are not ergonomic and introduce newer challenges such as neck fatigue, slower interactions with more errors, and reduced spatial awareness, while speech-based interfaces are challenging to use in loud environments, have high latency, and have difficulty interpreting different accents [35, 39]. For head-mounted displays (HMDs), mid-air interactions are a common input modality. However, these interactions can be uncomfortable and cause fatigue, often referred to as the "gorilla arm effect," [8–10, 16]. Additionally, regardless of XR interaction techniques, the corresponding virtual interfaces often tend to be flat-screen projections with point-and-click controls, largely inspired by desktop or mobile 2D user interfaces. These 2D-inspired approaches disregard

critical ergonomic factors, such as hand or finger reach and user pose [28].

Hands are our primary means of physical interaction, both with the world and with XR. Placing UI elements on or near the hand can leverage our familiarity and proprioception to enable the design of comfortable and easy-to-use interfaces [15, 51]. When designing XR interfaces, prior work has shown on-hand designs to be more subtle, less socially awkward [28], and less tiring [8, 27] than other forms of XR interactions. However, the scenario of a user grasping a physical object in one hand, with the interface reachable by the same hand is still unexplored. In our work, we focus on XR one-handed interfaces reachable by one hand while grasping an object, without extending the arm, and interactable with partial or complete hand or finger movements depending on the grasp. We refer to them as Grasp-Proximate User Interfaces (GPUIs).

The design of these interfaces, meant to be used by an occupied hand, requires a basic understanding of the hand-object interaction and how users manipulate their hands to engage with the interface. Creating one-handed interfaces for users engaged in physical tasks that involve tools or objects presents several challenges. Designers must consider how users grip objects (e.g., power grip for using a drill), which fingers have partial or full movement for interacting with the UI elements (e.g., thumb may have partial movement while holding a drill), and what are the reachable areas based on grasp. To address this, and assist designers in the creation of GPUIs, we defined a series of design factors and a design space that represents the hand-object interaction based on function, reachability, motion cost, and emerging boundaries. We then implement these design factors through a Unity simulation tool (GraVSim) to generate data that provides information about the hand segments' motion, reachability, and cost in the form of a three-dimensional space called Grasp Interaction Volume(GraV). To facilitate access to GraV data we share a dataset of GraVs that we generated using a combination of grasps (as defined by the taxonomy) and everyday objects in the YCB Affordance dataset [6].

We believe that GraV can assist designers in the development of grasp-proximate user interfaces by providing a better understanding of the hand-object coupled interaction through quantifiable data and tangible boundaries. GraV does not prescribe a GUI design; instead, it aims to inform design decisions by providing reach and interaction cost data. To explore this, we conducted a workshop and formative group interviews with XR designers from various backgrounds, where they created a series of grasp-proximate user interfaces for objects in a cooking scenario. The designers' explorations and key insights for the use of GraV in their workflows as well as future improvements of the GraVSim tool are presented as part of this work.

The proposed work builds on prior research on thumb-to-finger interactions [50, 54], microgestures [25, 42, 43, 45], on the hand interfaces [10] and explorations using physical objects as part of the interaction design [19, 24]. Yet, even with the increasing interest in making XR interfaces more practical for everyday real-world interactions, addressing the physical elements of these interactions is still a challenge for developers [1]. This is due to the lack of transferable design guidelines and technical solutions that provide quantifiable data to assist designers, particularly for grasp-proximate interfaces. The importance and value of these systems are emphasized by [9],

where they explore this approach by providing ergonomic data to optimally place mid-air interfaces within the proximity of the user body. We believe that the approach will help provide designers with resources to assist in breaching the gap between the design process and XR interfaces that can adapt physical tasks by facilitating the understanding of hand-object coupled interaction for grasp-proximate user interfaces. We present the following contributions:

- A set of design factors that represent the hand-object coupled interaction, and a design space comprised of the three fundamental variables always present in the design of one-handed grasp-proximate user interfaces (GPUIs) in XR.
- A downloadable Unity package (GraVSim) for designers and developers to create customized GraVs using our parameterized 3D hand model
- A GraV dataset, containing reachability and cost data as a point cloud, based on standard anthropometric values, grasp taxonomy, and common household objects.
- An evaluation of how GraV can be used to develop GPUIs through a workshop and formative interviews with XR designers. The workshop demonstrated how designers can approach the design of these types of interfaces using simulated quantifiable data, how they would incorporate this data in their workflows, and future recommendations to improve the simulation tool (GraVSim)

2 RELATED WORK

2.1 Design of Single-Handed Interfaces

To clarify how GPUIs and GraV relate to prior work, we present an overview of previous research on the design of interfaces with support to one-handed interactions accessed by hand motion. This space has been previously explored by Faleel et al. [11], who proposed a framework for hand-proximate user interfaces (HPUIs) which are virtual interfaces registered to the user’s hand or the space around it. Sharma et al. [43] elicited single-hand microgestures for handheld objects and proposed three categories of gestures according to their action location: In-Air, On-Body, and On-Object. Ergonomics of single-handed interactions have been studied, pointing out that factors such as interface orientation, size of handheld object, and arm pose can affect physical comfort and performance [9, 50, 55].

2.1.1 In-Air. In-air single-handed interactions use the available space around the hand to support interaction modalities such as XR interfaces [11, 26, 38, 55] and gestures [11, 42]. HPUIs that support in-air interactions can follow a layout that uses the space above the hand to show elements such as a panel of icons [11]. Kim et al. [26] compared in-air typing in VR with pseudo- and self-haptics alternatives, finding they support comparable typing performance but underperform in user experience and preference.

These interfaces can support in-air interactions with head-mounted displays in a single-hand-free setting, but their usability is compromised when the user grasps an object that obstructs finger motion. GraV mitigates this problem by providing information about the space reachable by the fingers while grasping an object. Designers of these types of interfaces must decide on the layout of elements, which is not a trivial decision given there is limited

information about the comfort of hand interactions during the design phase. GraV tackles this by providing joint rotation costs in a point cloud reachable by users’ fingers. Moreover, when the user’s hand diverges too much from the assumed design parameters, due to anatomic differences or motor disease, these interfaces can become unusable. GraV can be personalized for each user to reflect their individual interaction space, allowing designers to tailor their interfaces to the interaction space of a specific user.

In SoloFinger, researchers demonstrated a method to design microgestures that are robust to false activation based on the analysis of movement signatures, enabling users to perform gestures reliably while holding objects [42]. However, choosing among the possible microgesture options while considering different grasps and different objects can be a challenging task. GraV can be used by designers to identify the most suitable microgesture based on available movement space and pose costs.

Xu et al. [55] evaluated comfort metrics of interactions in the range of motion of the wrist and found physical comfort, pointing speed, and pointing accuracy to be higher for interfaces with fixed orientations. In more recent work, Evangelista et al. [9] proposed a toolkit to visualize the interaction cost of mid-air interactions, supporting the design of ergonomic XR interfaces. XRgonomics divides the mid-air interaction space into voxels that are accessible by a simplified arm structure using inverse kinematics. Different ergonomic costs can be assigned to the voxels. These projects can provide designers with valuable data about in-air interactions; however, they do not offer information about finger motion and do not consider motion obstructions caused by a grasped object or self-collision. To support the design of interfaces for hands grasping an object, we propose GraV, a volume that represents the space reached by the fingers of a grasping hand.

2.1.2 On-Hand and On-Body. On-hand single-handed interactions use the surface of the hand to enable interaction modalities such as XR interfaces [11, 26, 38, 53], on-body projected interfaces [15, 16, 34, 56], gestures [22, 45]. One key benefit of on-hand interfaces is the tactile feedback when the hand touches itself, commonly in finger-thumb contact. Another benefit of on-hand interfaces is the potential to leverage proprioception for eyes-free interaction [22].

HPUIs can support on-hand interactions by showing virtual interface elements on the hand and support several gestures and finger-thumb interactions [11, 38]. Virtual keyboard interfaces have been deployed as on-hand interactions where keys are pressed with finger-thumb gestures [26]. DigiTouch [53] is a glove-based input device that supports thumb-to-finger interactions targeted to head-mounted displays.

Another strategy to support on-hand interactions in single-handed interfaces is on-body projected interfaces. PALMbit [56] is an ONPI that uses a shoulder-worn projector and camera that uses the palm of the hand as an interface. SixthSense [34] is another example of ONPI powered by computer vision that allows the hands to be tracked for input. OmniTouch [15] has a similar shoulder-mounted projector that enables true touch inputs through depth sensing. DigitSpace [22] supports interaction with buttons, touchpads, and sliders based on thumb-to-finger motions. Soliman et al. [45] proposed a design space of thumb-to-finger microgestures accompanied by a recognition system powered by depth sensing.

These interfaces are effective for projected on-hand interactions in a setting where one hand is free. However, their performance becomes uncertain when the user holds an object that restricts finger movement. To address this issue, GraV offers insights into the reachable space for fingers. Designers of such interfaces face the challenge of determining the layout of elements, a non-trivial decision due to limited information on comfort during the design phase. GraV addresses this by providing joint rotation costs within a point cloud that can be accessed by users' fingers. Furthermore, when users' hand positions deviate significantly from the assumed design parameters, whether due to anatomical differences or motor disorders, these interfaces may become less effective. GraV offers the potential for personalization, allowing each user's individual interaction space to be considered.

2.1.3 On-Object. On-object single-handed interactions use the surface of the hand to enable interaction modalities such as XR interfaces [19, 21, 57], wearable controllers [2, 31, 46, 49], on-body projected interfaces [15, 16, 34, 56], and microgestures [42]. An additional benefit of on-object interfaces is the tactile feedback provided by the contact with the object.

The geometric alignment between a physical object and the virtual interface has been explored as one of the key factors that enable on-object in single-handed interfaces. In Annexing Reality [21], researchers matched virtual objects to physical proxies based on their shape similarity supporting haptic sensation for virtual objects. In UbiEdge [19] authors explore the use of edges, ubiquitous geometric features, to opportunistically place virtual interfaces.

While these interactions allow on-object interactions with opportunistic haptics, they overlook the ability of the user's hand to access different regions of the object when grasping with only one free hand. The grab interaction volumes we propose contain the surface of the object that is accessible by the user's hand while grasping an object. GraV focuses on the reachable space around a grasp, to help expand interface design possibilities while enabling the potential for opportunistic haptic feedback [13, 26] both from thumb-to-finger interactions and from finger-to-grasped object interactions.

Gripmarks [57] proposed the use of templates of grips, according to a proposed classification, and basic shapes of handheld objects to identify runtime opportunities for haptics. In a proof-of-concept demonstration, authors showcased the use of Gripmarks to create UIs aligned to an object surface that support tap, swipe, and gesture interactions. Even though Gripmarks supports grip-based opportunistic haptics in runtime, allowing users to interact with the system without letting go of the object in hand, it offers limited support during the UI design phase, primarily focusing on hand grip templates and object primitive shapes. GraV complements this approach by providing designers with finger motion data as an interaction volume point cloud that takes into consideration the limitations of finger range of motion (RoM) and arbitrary grasped object geometry.

Iconic wearable on-body interfaces include devices such as a one-handed keyboard attached to the palm [31], a wrist-bound touchpad [49], and glove and fingertip-based input systems [46]. GraV can support the design of on-body interfaces by making

designers aware of the space that can be reached by the fingers in a free-hand setting or while holding a physical device.

Trudeau et al. [50] studied motor performance during single-handed mobile phone use and found that smaller phones lead to better thumb performance in adduction-abduction movements.

2.2 Hand-Object Interaction Datasets

Hand-object interaction is a subject of interest in various fields including motion reconstruction, human-robot interaction, and action recognition [5, 12, 18]. To assist research in this area, multiple datasets have been constructed. ARTIC is a dataset of bimanual manipulation of articulated objects like laptops and scissors and contains contact information, 3D hand and object meshes [12]. The DexYCB dataset contains multiview RGB-D frames of 10 subjects grasping 20 objects and provides ground-truth hand and object 3D poses [5]. The ObMan dataset is a large-scale synthetic dataset of hand-object manipulation scenarios [18]. It comprises a total of 2772 3D models across eight daily life object categories. The grasps in this dataset were generated using GraspIt, a robotic grasp simulation software [33]. The GRAB dataset includes complete body motions recorded using motion capture markers from ten participants engaging with and grasping 51 different objects. It contains 3D hand and object positions, along with binary object contact maps [47]. While these datasets provide rich data for research on hand-object grasps, none of them contain grasp interaction volumes. The GraV dataset fills this gap and provides 367 grasp interaction volumes to support the design of GPUIs in XR. Our dataset builds on top of the YCB Affordance [6] dataset and contains 58 objects from the YCB object set [3] manually annotated with hand poses for each of the 33 types of grasps in the GRASP taxonomy [14]. Since not all the objects can be grasped with every grasp type, there are 367 valid combinations in total. Since the YCB dataset provides object poses and representative grasps, we use it as input data for the generation of our volume dataset, which contains 367 grasp interaction volumes generated from the valid object-grasp combinations in the YCB Affordance dataset.

Prior research has developed tools to support hand-object interaction data collection. One such tool is ARnnotate [40], which assists users in performing various hand poses while the system records 3D hand positions, 3D object bounding boxes, images, and additional metadata. Tools like ARnnotate could be utilized to gather data for expanding the GraV dataset we provide or for generating personalized GraV data in real-time.

3 INFLUENCING FACTORS AND DESIGN SPACE

In our daily lives, we often interact with physical objects with interfaces optimized for single-handed use. This single-handed interaction is a familiar and comfortable strategy that is already used in the design of common objects, such as TV remote controls, power drills, and computer mice. They capitalize on common interactions and poses that are familiar to the user to reduce the learning curve and provide comfort for prolonged use. Reachability can be optimized for supporting new functionality or expert users. For example, a gaming mouse has more buttons, but they are all reachable by the same fingers and similar grasp used for a non-gaming

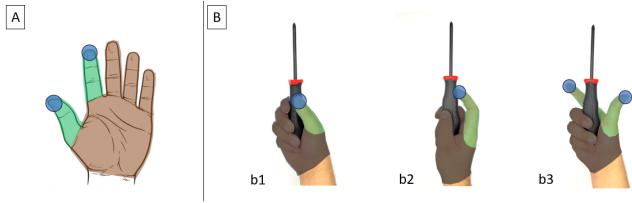


Figure 2: Hand Grasp and finger Interaction Parts. In (A), the brown region represents the part of the hand that is used for grip (i.e., the palm), the green represents the fingers that can move freely (index and thumb), and the blue dots represent the fingertips (both pads and nails) as the finger interaction parts. (b1) and (b2) represent the same grip but are adjusted to prioritize thumb and index finger mobility, respectively. (b3) shows both fingers are available to move while holding the screwdriver.

two-button mouse. Acknowledging this type of single-handed interaction (where the user grasps an object and interacts with an interface using the same hand) as a practical, comfortable, and familiar strategy for interacting with objects, we draw inspiration from this modality to formulate the factors that influence the design of interfaces reachable by a hand while grasping an object. In this work, we refer to this particular interface design scenario as GPUIs. We use the following definition of grasp and interchangeably use the word grip to mean the same thing. “A grasp is every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation.” [14].

In our exploration, we segmented the hand’s functions and its interaction with the grasped object to outline what parts of the hand could be used for interaction vs grasping (Section 3.1). Based on that we proceeded to identify the information needed to characterize the finger motion and its capacity for interaction (Section 3.2). Through this characterization, we identified two boundary conditions that occur due to the object’s intersection of the hand’s space and the finger’s reach. We also illustrate how the hand object itself could be used for opportunistic haptics. Finally, we present a three-dimensional design space based on the three fundamental variables that always need to be considered for the design of GPUIs. We provide these new factors based on our own observations and prior literature as a new framework to address the design of GPUIs.

3.1 Hand Segmentation: Grasp, Motion, and Interaction Parts

In this section, we present three categories we formulated to refer to distinct parts of the hand when grasping an object, with a specific focus on finger-based interactions. A fundamental aspect of the interaction between the hand and the object pertains to the equilibrium between ensuring grasp stability to prevent the device from falling or losing appropriate contact and allocating fingers for interacting with the intended interface. Everyday scenarios, such as smartphone usage prioritizing one-handed thumb access or using a video-game controller to balance the grasp and finger reach to incorporate speed and precision, exemplify this equilibrium. In addition, while the hand retains its hold on the object,



Figure 3: Hand Motion Tracking and Cost Evaluation. (a1) shows a 2D representation of the finger motion range for the index finger, while (a2) shows a 2D representation of the accessible interaction volume for the index fingertip. (b1) and (b2) represent a lower cost and a higher cost motion for the index finger, respectively.

depending on the intended interaction, the grasp can be relaxed to allow for more reach by one or more fingers. This mimics the use of one-handed tools, such as adjusting the velocity or direction of a power drill. How a person grasps a physical object depends on both the object’s characteristics (e.g., weight and shape) and the nature of the task being performed. Interactions with an interface add a new dimension to hand-object interactions where the UI elements and corresponding finger-based interaction techniques need to be considered. The following components divide the hand into three functional categories:

Grasping Elements. The parts of the hand that are used for grasping an object resulting in their restricted motion. They change depending on the grasp type and the object being grasped. The brown shaded part of the hand presented in Figure 2 part A provides an example of this for a hand using 3 fingers for grasping.

Unconstrained Elements. The parts of the grasping hand that can move without compromising the grip equilibrium and the capability to perform the intended physical task with the grasped object. In Figure 2 part A, this is represented by the fingers shaded green.

Interaction Elements. The parts of the hand and finger segments that can be used to interact with the virtual interface. Depending on the intended interaction modality and RoM, different segments of different fingers can be used to interact with the UI elements (e.g., knuckle, fingerpad, nail). Based on comfort findings in prior work [22], we chose the fingertips, including both the bottom and top of the fingertips (i.e., fingerpad and nail), as a point input in our work. A representation of this is provided in Figure 2 part A shown as blue circles on the fingertips.

When designing a GPU for finger interaction, a designer would first need to determine the grasping elements followed by identifying the elements that can move and their RoM. Once the available fingers are identified, designers need to select their preferred finger segment(s) to interact with the interface elements or track for gesture interactions. Figure 2 part B represents how a user can prioritize the use of either the thumb (b1) or the index finger (b2) while holding a screwdriver. Part (b3) in the same figure, represents the user choosing a grip that allows them to use both the index and thumb and their fingertips as input interaction parts.

3.2 Motion, Reach, and Displacement Cost

Here we focus on the finger's capacity to move and interact with the interface. To assess the interaction possibilities and limitations, we segment finger actions into three factors. The Finger Motion Range represents the motion capacity with respect to the grasped object, the Accessible Interaction Volume represents the available interaction space, and the Displacement Cost shows the effort needed to traverse the Accessible Interaction Volume, and the finger Displacement Cost. Each of these factors represents a concrete set of ranges that can be used to determine potential interactions.

Finger Motion Range. The total motion space that is available to unconstrained fingers (i.e., the maximum finger RoM) when an object is grasped. The RoM gets obstructed by the object and the grasp type due to self-collision. Figure 3 part A:a1, presents an example of a finger that cannot go lower due to the surface of the screwdriver's handle. This can also be evaluated for other parts of the hand.

Reachable Interaction Volume. The motion space that can be accessed by the *Interaction Elements* when holding an object. Similar to the motion range, the interaction volume changes depending on the object and grip type. It also changes depending on the selected finger segments for interaction (e.g., fingertip vs. the second digit) as seen in Figure 3 part A:a2. In that example, the blue dots represent the tracking of the fingertip as the interaction part, and the purple shade represents the *Reachable Interaction Volume*.

Displacement Cost. The cost of the fingertip to reach any point in the interaction space from the resting position of the grip. Different metrics can be used to estimate the cost using factors like joint angular displacement, muscle strain, and subjective discomfort levels. In our work, we focus on joint rotation costs. Figure 3 part B shows how reaching a higher point in space with a straight index finger (b1) might be less costly than traversing the tool surface by curling the finger (b2).

To effectively design a GPU, it is imperative to consider the motion limitations of the hand and the reachable interaction volume when grasping an object, because the primary design element of GPUs is to be reachable while grasping an object. The Displacement Cost can help provide a reference map of areas in the interaction space that can be used or should be avoided by developers.

3.3 Boundaries

In this section, we define the hand as a system possessing a maximum RoM in its unobstructed or free-hand state (Figure 4 part A), and a constrained RoM when grasping an object or in the grip state. In the free-hand state, both the palm and fingers can execute unrestricted movements (e.g., flexion, extension), with self-collision being the main limitation. However, when the hand grasps different objects, this freedom is impeded, diminishing hand motion and finger reachability. We outline two boundaries that emerge from the free- and grip-states, namely, an object boundary, which is caused by the intersecting volume of the object being grasped, and the motion boundary, which represents the maximum outward reach of the fingers. While our main focus in this work is on the fingers, other parts of the hand can be included in the evaluation of the boundaries.

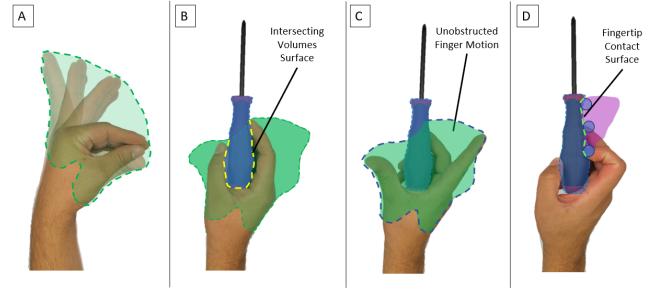


Figure 4: Object and motion boundaries. (A) shows a free-hand range of motion, and (B) shows the object boundary that is formed when an object's volume intersects with the hand's free-motion volume. (C) represents the motion boundary for both the index finger and thumb while holding a screwdriver with no external obstruction. (D) Shows the opportunistic surface haptic available for the index finger.

3.3.1 Object and Motion Boundaries. When an object is grasped, it intrudes upon the mobility space of the hand, constraining the range of motion of the fingers. It also restricts the hand's capacity to adjust the grip based on factors such as weight, size, and intended use, among others.

Object Boundary. The part of the object that intersects the RoM of the free hand creates a boundary surface between the object and the RoM. This results in diminishing the available free-hand RoM as seen in Figure 4 part B.

Motion Boundary. This represents the surface of the outer trajectory of the finger's RoM that is not obstructed by the object (or self-collision of the fingers). Essentially, for each of the fingers, as they traverse the volume they can move in, the external boundary is the surface that represents their maximum unobstructed reach as presented in Figure 4 parts C.

These two boundaries represent a dual envelope condition that surrounds the hand when grasping an object. The object boundary represents the motion limitations with respect to the object and the motion boundary represents the furthest unobstructed fingertip reach point across its motion space. The motion boundary, even though it is primarily dependent on finger reach, can be reduced by intersecting environment elements (e.g., using a screwdriver inside a tight corner in a cabinet). Knowing the environmental elements, that can intersect the *Motion Boundary*, can be particularly useful when estimating the available interaction space. It is worth noting that in cases where the grasped object goes around the hand or the fingers (i.e., holding a pitcher), the pitcher would be considered as an object boundary and not an environment element. Section 4 expands on this with the Unity Tool workflow to block regions of the object that should not be used for the placement of interface elements.

3.3.2 Opportunistic Surface Haptics. When we look at the intersection of the object boundary and the interaction space, we get a sub-section of the object surface that can be reached by the input region of the finger (i.e., the fingertip) Figure 4. These surfaces offer a unique opportunity to provide passive haptic feedback as allowed

by the grip and the task, in addition to thumb-to-finger haptics [10, 13].

3.4 GPU Design Space

Lastly, we construct a GPU design space with three axes: grip, object, and hand (Figure 5). These fundamental variables always need to be considered when defining the hand-object coupled interaction for GPU design.

Grip Types. Despite individual variations, most grips fall under one of the types presented in the grip taxonomy [14]. The GPU designer can begin by approximating the user’s grip, using the taxonomy as a reference (Section 4). Figure 5 part A provides an example of this.

Grasped Objects. Another axis is the grasped objects. Designers can traverse the space keeping the grip type and the hand as constants, and changing the grasped objects as seen in Figure 5 part B.

Hand Type. Hands are not the same. They vary in size, motion range, strength, and comfort preferences across users. Hands change over time, temporarily or long-term due to injury, disease, or fatigue. Variations can also be as simple as considering a right vs. a left hand. Figure 5 part C shows an example where both the grasp type and the object are constant but because the thumb has an injury the thumb RoM is greatly reduced.

The last factor that can be adjusted is the XR application. The interface layout needs to take into account the grasp, the user’s hand, and the grasped object, but the designer can choose how to balance these elements. For example, they could keep the grasp and object constant while adjusting the application and related UI elements to design an ergonomic and comfortable user experience with UI elements that are always reachable regardless of their number or type.

4 GRASP INTERACTION VOLUME

RoM, or range of motion, is crucial for understanding the potential movement and reachability of the grasping hand. It is defined by the full extent of each finger’s movements such as flexion, extension, adduction, abduction, pronation, and supination [4]. Since each joint, such as the wrist, knuckles, and fingers, has a limited range of movement, those limits form a boundary for the overall RoM. The motion boundary and positions of these joints relative to each other and to the grasped object form a complex set of constraints, which have implications for the design of ergonomic applications in various fields such as industrial design, robotics, and rehabilitation [29]. Techniques based on a joint’s RoM also have the potential to assist designers of XR user interfaces in making data-driven layout decisions as demonstrated in XRgonomics [9] by the mid-air arm interaction space or by a motion range semi-sphere attached to the wrist [55].

GraV enables access to the *Reachable Interaction Volume* (defined in Section 3.2) spanned by the hand’s RoM that can help designers create reachable and comfortable interfaces. The shape of the volume is restricted by the surface of the grasped object, self-collision of the fingers, and the limits of the hand RoM. In addition to the 3D volume, GraV encodes the joint rotation cost with respect to

an initial pose and is visualized as a color-coded map in the point cloud. We built a GraV dataset and a simulation tool in Unity to make it easy for designers and developers to access and customize GraV’s for their user scenarios. Besides using the volumes with the grasp types and objects we provide in the GraV dataset, designers can simulate their own volumes by setting customized joint orientations and object surfaces. These input parameters can be collected through methods such hand tracking¹ and photogrammetry², hand-object interaction datasets [5, 6, 12, 18], or predicting hand grasps [33, 48] over a dataset of object 3D meshes [3], among other approaches.

4.1 Hand and Grasp Parameters

The anatomy of the hand is intricate and can differ between populations and change over an individual’s lifetime due to factors like growth, injury, or disease. Our simulation technique takes a range of input parameters that designers can modify to represent a variety of hand types. Designers can specify the relative positions of joints in a hierarchy, to control essential anthropometric measurements such as hand length, breadth, and maximum spread. Furthermore, designers can define the RoM for each joint by setting minimum and maximum angles around each axis, making it possible to simulate interaction volumes for hands with limited motion or hyper-extension. Properly defined RoM values can prevent fingers from bending in unnatural ways. Hand thickness can also be parameterized along each inter-joint segment, which allows designers to simulate various levels of contact between the finger pads and the grasped object. While we provide a large set of interaction volumes in our GraV dataset, designers can simulate their own volumes by customizing anthropometric parameters in our Unity simulation package, which we will make available along with our dataset upon publication (Section 5). Designers can also choose to obtain joint hierarchy, positions, RoM, and hand thickness from external sources such as anthropometric databases [4], or neural networks-based hand models [30, 41].

Consistent rotation orientations are critical for accurate simulations. We keep the same 20 joint hierarchy as defined in the normative model of hand [4] while modifying the joint orientations to match Unity’s left-handed coordinate system. The wrist is linked to the fingers along the joints MP (Metacarpophalangeal), PIP (Proximal Interphalangeal), and DIP (Distal Interphalangeal) in that order. The wrist is connected to the thumb along the joints CMC (Carpometacarpal), MCP (Metacarpophalangeal), and IP (Interphalangeal) in that order.

We adopt the following conventions for joint orientations, also depicted in Figure 6. The forward direction (positive Z) of each joint is defined by the normalized vector pointing from the parent joint to the child joint. In the case of the wrist joint, the forward direction is determined by the normalized vector pointing from the wrist joint to the centroid of the hand. The right direction (positive X) of the fingers’ joints is defined by the vector pointing from the little MCP to the index MCP. The right direction of the thumb joints is defined by the cross-product between the vector pointing from the CMC to the MP and the vector pointing from the MCP to the

¹<https://developers.google.com/mediapipe>

²<https://poly.cam/>

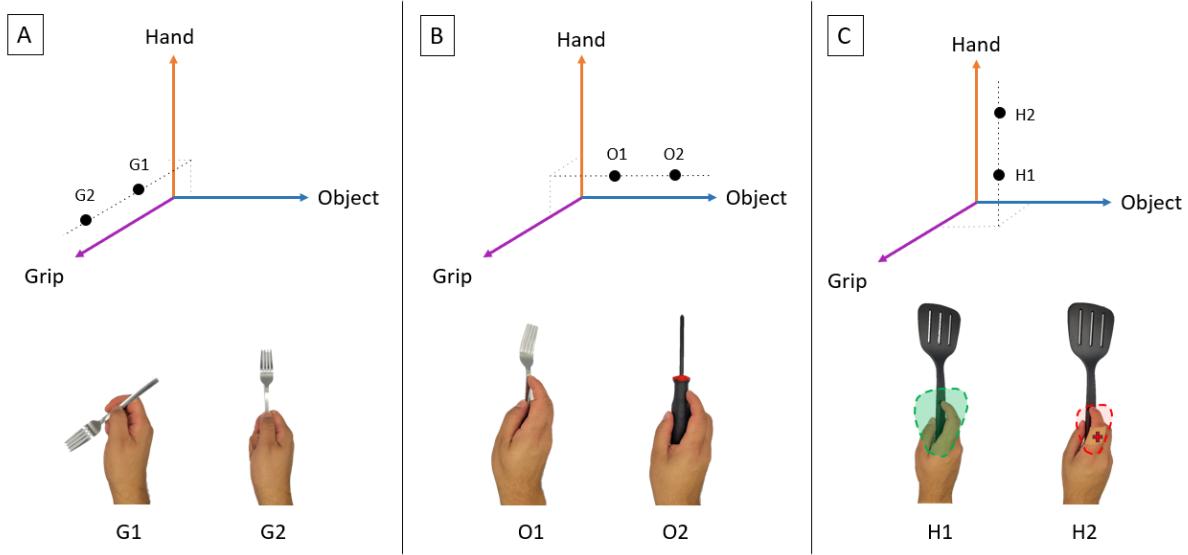


Figure 5: Design space for GPUIs. (A) represents a change in grasp type keeping the user's hand and object constant. (B) represents a change in an object while keeping the other two axes constant. (C) represents the same object and grasp type for the same user but in condition H2 the thumb has an injury that reduces its range of motion.

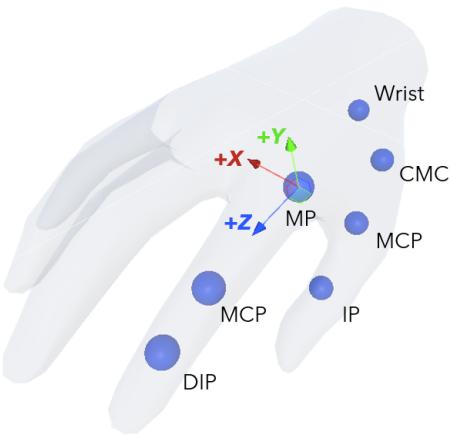


Figure 6: Procedural hand generated in the simulation process from anthropometric parameters. The forward direction points in the direction of the index. The right direction points from the index MP to the little finger MP. Positive rotations around the X-axis are counterclockwise.

IP. We define these conventions to ensure a consistent reference for rotation directions across all the joints in our simulations.

Since the hand cannot intersect with the object or with itself, the finger self-collisions, finger-palm collisions, and finger-object collisions are discarded in the generation of the reachable interaction volume. To cover a wide variety of grasp configurations, our technique allows designers to input an object's surface geometry as a 3D mesh, an initial grasp pose defined by joint rotations, in addition to hand parameters as mentioned in Section 4.1.

4.2 GraVSim

Based on the design factors presented in Section 3, we developed a tool in Unity that can simulate the motion of a grasping hand and generate GraV in the format of 3D point clouds color-coded by motion cost. To create GraV, GraVSim needs an object's surface, a joint hierarchy, and joint range of motion, initial rotations, and positions. GraVSim generates the *Reachable Interaction Volume* by independently simulating the motion of each fingertip in the grasping hand. Selecting the fingertips as the *Interaction Element*, the interaction volume of a hand includes all the points in the volume that can be reached by any movable fingertip, and conversely, any point within the interaction volume can be reached by those fingertips. Any positions where any part of the finger that intersects the object's surface are excluded from this volume.

The simulation method employed by GraVSim is based on a Flood fill algorithm, which is behind paint bucket tools in many graphic editors. Flood fill finds connected regions adjacent to an initial pixel of an image limited by sharp boundaries [44]. Similarly, we use the flood fill algorithm to explore the hand RoM by rotating a joint in discrete angular steps to iterate through adjacent rotations. In our case, the filling process stops at the boundaries of the RoM or when the finger collides with a grasped object or against the hand. Figure 7a shows the stop conditions. Our simulation applies the flood fill algorithm for each finger independently starting from the joint closer to the fingertips (DIP of the fingers and IP of the thumb) until the root of the finger (MCP of the fingers and CMC of the thumb). After completing the flood fill in an initial joint, the simulation proceeds to the parent of that joint, taking a single filling step, i.e., just exploring the immediate neighborhood of that joint's current rotation. After this single step, the simulation executes an entire flood fill procedure for the child of that joint once again.

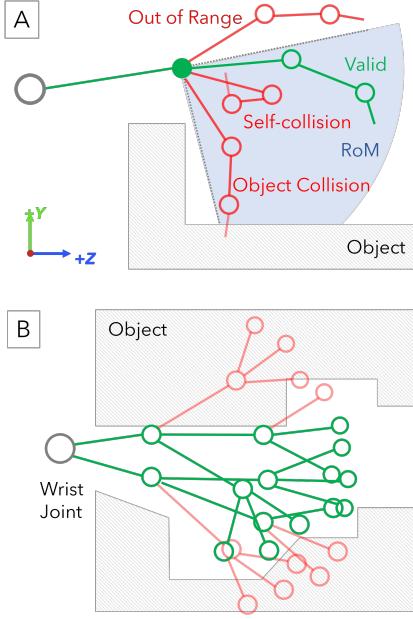


Figure 7: GraV Simulation Process. (A) the simulation at the joint level stops when the fingertip either: goes beyond the limits of the RoM, collides against the hand, or collides with an object. (B) The execution expands along the joint hierarchy forming a tree-like structure.

This exhaustive hierarchical execution allows us to traverse the adjacent valid joint configurations with respect to the initial hand pose and find a connected volume where the fingertip can move unobstructed. We call this volume GraV. Figure 7b shows a diagram of the execution tree process reaching valid joint configurations.

At each step, the simulation records the position of the fingertip and a cost metric, resulting in a 4D point cloud. We adopted the total joint rotation cost, calculated as the sum of the joint angles between the current pose and the initial pose from all the finger joints. To manage computational complexity, we traverse the RoM of a joint in discrete angular steps defined by a minimum angular step parameter. Smaller angular steps result in denser volumes and longer execution times, while larger angular steps result in sparser volumes and shorter execution times.

GraVSim is implemented in Unity, which allows designers to import and manipulate its input parameters directly in the same ecosystem as other XR SDKs they use for building XR experiences. Additionally, we offer a command-line interface that facilitates the manipulation of volume data and conversion into CSV, PLY, and OBJ formats for export into other tools.

4.3 GraV Dataset

To support designers creating GUIs and demonstrate the feasibility of our simulation procedure, we generated a GraV dataset containing 367 grasp interaction volumes. We build our dataset using as input parameters grasp poses and objects from the YCB Affordance dataset [6], which is a publicly available dataset that

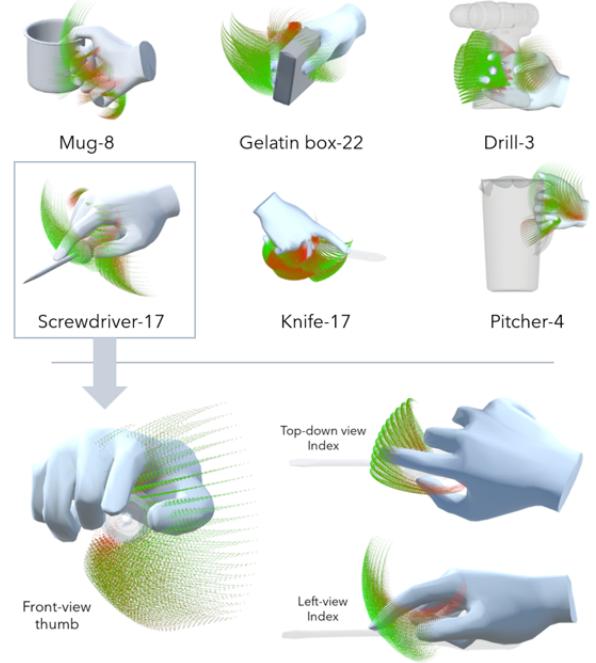


Figure 8: GraV Dataset contains 367 grasp interaction volumes from feasible combinations of 33 grasp types and 58 objects available in YCB Affordance [6]. GraV instances are labeled according to object type from the YCB object set [3] and a number from 1 to 33 indicating the grasp type according to the GRASP taxonomy [14].

contains each of the 33 grasp types proposed in the GRASP taxonomy [14] manually annotated over 58 objects from the YCB object set [3]. We provide GraV's in the form of CSV, PLY, and OBJ files labeled with object and grasp types. The minimum angular step in the simulations to generate the GraV dataset was 5 degrees. Figure 8 shows examples of grasp interaction volumes that we generated. The annotated grasp poses in the YCB Affordance dataset are provided as MANO representations [41], a neural network-based hand model capable of representing a wide range of hand types and poses. We access the joints provided by the MANO representation to initialize the joint positions in our simulations to create the interaction volumes. The joint hierarchy is defined according to the normative model of the human hand [4]. We calculate the joint orientations based on the joint positions to match the conventions presented in Section 4.2. We initialize the hand joint RoM in our simulations with the values provided in the anthropometric dataset [4]. We initialize the thickness of the hand in our simulation with the median hand thickness of the male American population [4]. To reduce parameter complexity, hand thickness is assumed to be constant over the entire hand.

4.4 Designer Workflows

As shown in Figure 9, GraV dataset and the Unity Simulation package can both be used (and modified) by the designer in a variety

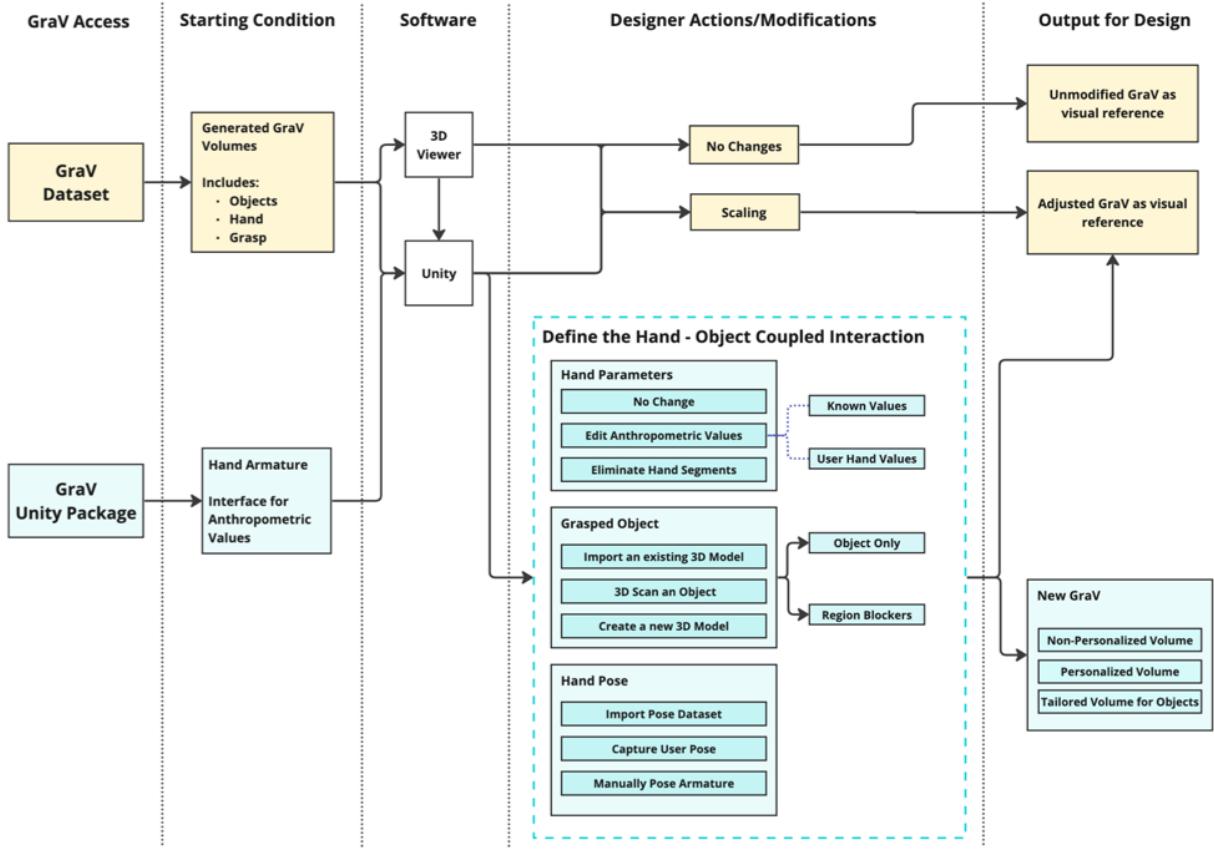


Figure 9: Two different pathways for designers to access GraVs: pre-created GraVs from our dataset (yellow) and the Unity Simulation Tool to create new GraVs (blue).

of ways to provide volume data to inform UI placement. Designers can import the dataset or individual volumes from the dataset into Unity or other compatible software to access the 3D representation of the hand-object coupled interaction and the reachable volume. They can use the volume data as-is to assess reachability or scale it to explore UI designs for different hand sizes. The Unity Simulation package includes a poseable hand armature and an interface to modify hand anthropometric parameters. In this workflow, designers can freely define their own hand parameters, import a 3D object model, and choose a grasp that best represents their user scenario. In addition, they can designate certain areas on the grasped object or surrounding space as Region Blockers. These blockers aim to guide designs toward specific reachable regions, to either constrain movement or prevent access to hazardous parts of the object (such as hot regions or blades) or places already equipped with physical interfaces (like buttons, switches, or sensors).

5 DESIGN WORKSHOP

To assess GraV's applicability, we conducted a design workshop, individual questionnaires, and group discussions with XR designers, developers, and researchers. It was approved by our local IRB (protocol 7-22-0512) and participants provided informed consent. Our study aimed to:

- Explore how XR developers could approach the design of grasp-proximate user interfaces supporting physical tasks.
- Understand how GraV data (reachability and displacement cost) can inform design decisions, how designers would incorporate it in their workflow, and recommendations to improve GraVSim.

The design constraints were derived from presented design factors (Section 3) while the explored cases traversed the axes of the design space. To prevent any biases during the exploration of the designers, the design factors (Section 3) were omitted from the workshop, and only the design space (Section 3.4) was mentioned to introduce the design scenarios.

5.1 Participants

We recruited nine participants (5 males and 4 females; age range: 20-34). All participants were familiar with XR and UI design as they were either professional software developers or VR/MR researchers with 1-5 years of VR/MR experience. All participants mentioned Unity as one of their primary design tools. Some of the other tools they had experience using were Unreal Engine, Snapchat Lens Studio, Facebook Spark, AR Core, and AR Kit. Previous work included the design of co-located social games in AR, interfaces for augmenting spatial-awareness MR, real-time MR task guidance

system design, open space navigation with AR assistance, location-based immersive performance design, and interfaces to facilitate interaction between handicrafts and digital fabrication.

5.2 Apparatus

The user study was conducted in two group sessions. Due to geographical location and time zone differences, we opted for a hybrid approach combining in-person and video conferencing using Zoom. To support seamless communication between the remote and local participants, we projected the virtual participants onto a large screen. We used the Meeting OWL³ three with a 360-degree camera, speakers, and microphone capable of bringing into focus one or more speaking participants. For the study design tasks, participants used Unity as their design environment and were provided study assets to download at the start of the study. The 3D assets included the objects, hand model with the selected grasp, and generated reachability point clouds with color-coded displacement costs (GraVs). We provided physical props for the designers who preferred to interact with the physical objects while designing. For the remote participants, a list of the objects was sent before the workshop with minimal specifications to prevent bias. Since the items were common household kitchen items, they were instructed to acquire equivalent items in case they could not find exact replicas. Participants were informed during the workshop that using these objects was optional, allowing them to reference them based on their preferences.

5.3 Evaluation Design and Procedure

5.3.1 Workshop Design. We conducted two 3-hour workshop sessions, with 5 and 4 participants each. Our study consisted of a study setup and 5 phases as summarized in 10. To facilitate the exploration of the GPUUI design process, our study consisted of three UI design tasks. An introductory design task to introduce the design scenario and constraints of one-handed UI design for the physical task (Phase 1), a design task without GraV data (Phase 2) to explore the GPUUI design space (presented 3), and finally with GraV data we explore how the provided data could be used by designers in approaching the design of GPUUIs. For the *With-GraV* and *Without GraV* design phases, to provide a set of constraints that also traversed the three axes of the design space presented in Section 3.4, we randomly split the participants into three groups. Participants in these groups were assigned different interface design conditions where either the grasp, the object, or the hand type changed (e.g., injured finger). For the UI prototype, participants focused on designing a simple, widget-based UI that could be used to interact with an AR task guidance system for cooking. The intention of the design tasks was not to assess the appropriateness of their UI designs but to use the design tasks as constrained scenarios for the participants to explore the intricacies of creating GPUUIs.

The workshop's intent was not to evaluate a system but to explore how designers could approach the GPUUI design space and use the GraV data in their design workflows while maintaining a certain level of agency. To capitalize on the participant's initial impressions of GraV and their preliminary approach to incorporating reachability information into their workflow, we decided to focus

³<https://owllabs.com/products/meeting-owl-3>

on a short exploration. To maintain a minimum iterative design workflow, the workshop included multiple design tasks that built upon previous tasks, creating a continuous effort throughout the study. Participants transitioned from the baseline (Without-GraV) condition to the alternative condition (With-GraV), allowing for a direct comparison of their experiences in designing UIs using both processes. The decision to maintain a consistent order of conditions is based on considerations aimed at planning for ecological validity [17]. This approach aligns with real-world scenarios where individuals would be familiar with a design process, similar to the baseline, before engaging with GraV. Additionally, we identify the potential for asymmetric skill transfer between conditions, where exposure to reach and cost information from GraV may influence behavior in the baseline condition, while the baseline does not yield new insights for the alternative [32]. Lastly, we anticipated minimal additional learning about Unity, the underlying platform of the study, given participants' extensive prior experience with the tool.

5.3.2 Design Scenario Context. To provide context about the use of the objects and the target user interactions, the provided scenario was a novice chef using an AR task guidance application that provides them with instructions on how to accomplish a cooking task. The chef needs to be alert at all times and keep their eyes on the cooking task while interacting with the interface (without putting down the tool) to follow the instructions, hence the need for GPUUIs. All design tasks consisted of prototyping a UI for the provided cooking scenario. To simplify the task, all participants were provided with 4 UI widgets for navigating forward and back through task instructions, viewing the full recipe, and watching a video for help with any step. These widgets were used in all three design tasks.

5.3.3 Data Collection and Analysis. We collected individual questionnaires and conducted researcher-led group discussions. The combination of the design tasks and both individual forms and group discussion questions were chosen to help elucidate not only the challenges that participants encountered but also their underlying assumptions and the design strategies they employed to overcome these challenges. Participants uploaded their UI designs at the end of each phase. For the questionnaire, open-ended questions focused on their thought process, design justifications, and any insights. For the discussions, a researcher guided the conversation with open-ended questions, and all participants were encouraged to respond and build upon each other's comments by expressing similar or contrasting experiences during their design tasks. The focus of the group discussion was on the challenges, assumptions, strategies, ergonomic assessments, and potential uses of GraV (Phase 4) they identified during the study. We analyzed both the prototypes and feedback in a two-step process. First, two researchers analyzed the prototypes independently to identify patterns. Following that, the researchers analyzed the feedback to categorize themes correlated with the UI design prototypes, resulting in six main clusters. Section 6 provides more details about the resulting clusters and insights from the designers.

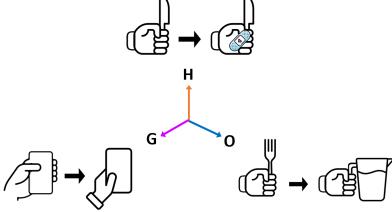
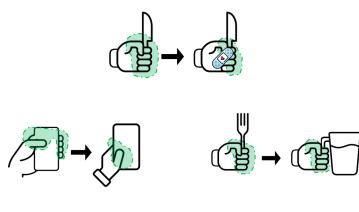
Study Phase	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Task	 <p>Introduction to the problem space and the cooking scenario, introductory design task, questionnaire</p>	 <p>Intro to the GPU design space, 3 variables, design task w/o GraV, questionnaire</p>	 <p>Researcher-led group discussion on Phase 1 and 2 design tasks</p>	 <p>Intro to GraV, design task w/GraV, Phase 2 design variables, questionnaires</p>	 <p>Final discussion to reflect on the impact of GraV on the design of UIs</p>
Apparatus	 <p>PC to run Unity and answer questionnaires, model of hand holding object, UI button assets</p>	 <p>3D Models of the objects, grasps and hands (injured vs not) types, w/o GraV</p>		 <p>Same 3D Models of the objects. Grasps and hand (injured vs not) types, w/GraV</p>	
Output	<p>UI prototype of hand grasping an object, questionnaire data</p>	<p>2 UI prototypes each exploring the 3 design space variables w/o GraV</p>	<p>Recording of group discussion on challenges, assumptions and design strategies</p>	<p>2 GPU prototypes exploring the 3 design space variables w/GraV</p>	<p>Recording of group discussion on challenges, assumptions and design strategies</p>
Duration	Approx. 40min	Approx. 40min	Approx. 30min	Approx. 40min	Approx. 30min

Figure 10: Overview of the GraV formative study procedure consisting of three design tasks (Phases 1, 2, and 4) and two group discussions (Phases 3 and 5), where XR designers created 3D UIs with- and without-GraVdata.

5.4 Design Tasks

The design tasks were presented during Phases 1, 2, and 4 of the study (Figure 10). In this section, we present the design constraints for each of the design tasks as well as the objects, grasp types (based on the GRASP taxonomy [14]), and hand types.

5.4.1 Phase 1: Introductory Design Task. For this task, participants were provided with a 3D model of a hand grasping a knife Figure 11. This task served as an initial exploration of the constraints related to GPU design and as a clarification exercise on doing the design task and uploading their build and design documentation.

Participants were provided with the following specific design constraints:

- The UI should be reachable by the fingers holding the object, without dropping it or requiring the other hand.
- No gaze, gesture, or voice commands.
- The interface can only be created with the provided UI widgets.
- Each UI widget must be reachable by at least one finger, but multiple-finger access is acceptable.

5.4.2 Phase 2: Without-GraV Design Task. For this task, the axes of the design space (grasp type, object, and hand type) were introduced. Participants were randomly split into three groups and

each group was provided with two 3D models representing their group's respective axis. The object group had two different objects (a water pitcher, and a fork) with the same grasp and hand type. The grasp group had two different grasps (grasp 22 and grasp 31 from the taxonomy [14]) with the same hand and object. The hand group had two different hand types (injured finger, uninjured hand) with the same grasp and object types. For the hand condition, the model was a hand holding a knife using a power grip, half of the index finger was removed from the second model of the hand to represent an injury.

The design constraints for this task remained the same as the introductory design task with the following additions:

- The user can reach the UI elements at all times without extending their arm.
- One prototype for each assigned 3D model representing a change in grip type, object, or hand type.

5.4.3 Phase 4: With-GraV Design Task. For this task, we introduced the participants to GraV and explained how the point cloud was generated and the cost metric calculated to colorize the point cloud. To allow for a more open exploration participants were told to use their own discretion and use the information provided by GraV, based on their design criteria, instead of simply placing the UI widgets within the volume.

The design constraints for this task also remained the same as the introductory design and phase 2 tasks with the addition of the use of GraV along with the same 3D models provided for the Phase 2 baseline design task.

6 RESULTS

6.1 Interfaces Designed by Participants

Each of the 9 designers in the workshop produced one UI for the introductory task and two UIs each in Phases 2 and 4, for a total of 36 interfaces and 144 button placements. This section provides an overview of the evaluation of their UIs and results obtained from these qualitative inquiries in the workshop.

Introductory Designs. The UI designs for the introductory task (Phase 1) were clustered into three groups: (1) opportunistic haptics - designs where UI elements are aligned with the held object's surface enabling passive haptic feedback; (2) hand-proximate - designs where UI elements are intended to be reachable by the fingers; and (3) mid-air - designs where UI elements are not within finger reach and would require arm motion to be reached. Figure 11 shows examples from each category. The point grip on the knife blade (often used by sushi chefs⁴), was taken advantage of by designer P1-G (based on their blade handling experience) to place all UI buttons along the dull edge of the blade (Figure 11).

Design Variations. We clustered the designs from Phases 2 and 4 into three groups, presented in Figure 12. The clustering is based on how the designs changed from those created in Phase 2, before using GraV, to designs created in Phase 4, after using GraV. The three categories are, (1) Layout Change - groups together the designs with changes in the UI element's position, rotation, or scale to fit the GraV; (2) Interaction Change - groups the designs that changed to allow interaction with fingers and motions not previously considered; and (3) Goal Change - groups the designs that changed to support a design goal such as reducing accidental interactions referred to as the Midas Touch problem [23] or favoring interactions with functionalities of assumed priority.

6.2 Feedback from Designers

Through the individual questionnaires and the group discussions, we aimed to learn about participants' prior knowledge and design processes and how they might apply those to one-handed UI design, the expected challenges for one-handed interactions, and potential design strategies to overcome the GPUUI constraints. After presenting everyday scenarios, in the introduction design task, such as using a screwdriver or drill to assemble furniture, participants expressed that they could relate to the disruption flow and fatigue when interacting with mid-air interfaces. They also acknowledge the challenge of one-handed interaction design, particularly when compared with more familiar approaches like mid-air, on-body, and on-object interactions requiring two hands. Furthermore, none had previously addressed the grasp-proximate one-handed scenario we presented, and they were not aware of existing tools, data, or strategies that could help with the design of GPUUIs. These insights

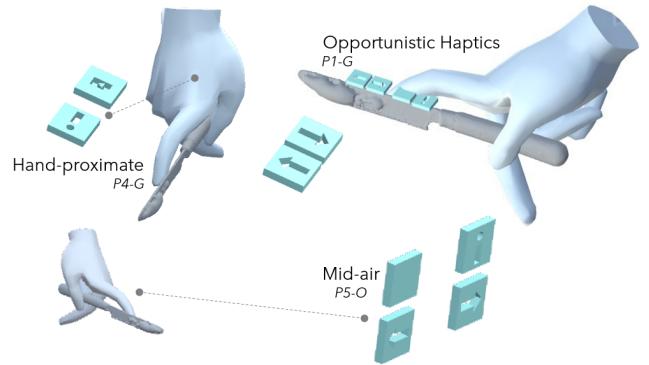


Figure 11: Designs created by participants in the introductory task for point grip knife blade 3D model from the YCB dataset [3], showcasing each of the three identified categories. The “Opportunistic Haptics” prototypes aligned buttons with the knife’s surface, enabling passive opportunistic haptic feedback (P1). “Hand-proximate” prototypes featured buttons intended to be reached by the fingers (P4). In the “Mid-air” prototypes buttons were accessible by arm motion and outside finger reach (P5).

reinforce the key challenges in designing XR for physical interactions, as previously presented by Ashtari et al [1].

6.2.1 Without-Grav. Below, we categorize participants' input into three categories, derived from the initial discussion (phase 3) and individual questionnaires, centered around tasks formulated without GraV.

Expressed Challenges. Participants expressed that the limited space available for the interface layout around the grasping hand was a challenge. Among the factors that limited the layout space, they identified the area the user could reach with their fingers, the surface of the object covered by the grasping hand, the grasped object itself and anticipated surrounding obstructions. Participants manifested concerns about obstruction between virtual and physical elements, e.g., interface elements occluding the field of view needed for performing the physical task, and physical elements blocking the interaction with interface elements. *“Different environment configurations would impose different ergonomic constraints (e.g., something is blocking the buttons in front of you).”* - P5. Another commonly expressed concern was that reflex (mirror) hand movement could cause accidental interactions with the interface elements and result in undesired physical consequences, such as spilling the contents of a pitcher as the user reaches forward with the thumb [26, 35]. P3 expressed this challenge by saying, *“It is important to know when you are moving a finger...how that impacts other fingers’ movement. Because if I am moving my ring finger...this one moves automatically (showing the little finger) so I could misclick a button.”*

Adopted Task and User Assumptions. Participants presented a series of adopted assumptions during the design process. Some revealed that they assumed the user’s hand had a certain size and scaled the interface elements accordingly. Some assumed that

⁴https://web-japan.org/trends/11_tech-life/tec202211_japanese-knives.html

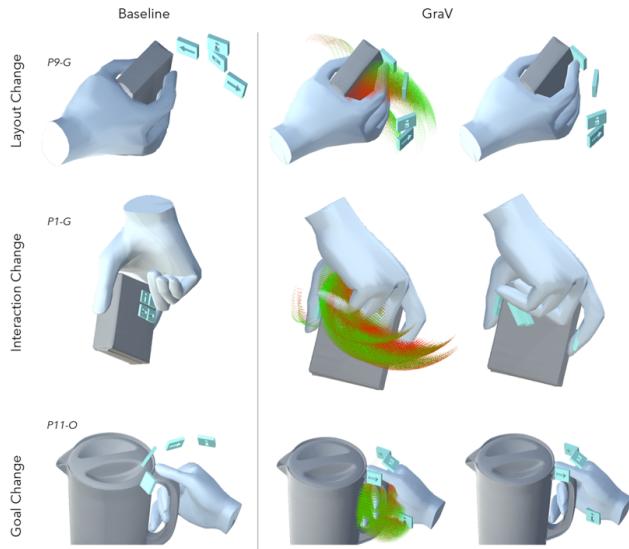


Figure 12: Designs created by participants using the three variables of hand, object, and grasp type in Phase 4 with GraV. In the Layout Change row, buttons in front of the box were moved to fit within the interaction volume (P9, grasp group). In the Interaction Change row, buttons were moved from the surface of the box to the area underneath the palm (P1, grasp group). In the “Goal Change” row, a UI widget was placed near the wrist to avoid accidental interactions (P11, object group). The leftmost column shows the designs Without-GraV and the two columns on the right show the updated versions of the designs in the left column created With-GraV.

certain fingers would always be grasping the object and therefore designed primarily for the index finger and the thumb, which they felt more familiar with. Participants also shared assumptions they adopted related to the priority of elements in the interface and functionalities of the application.

Implemented Design Strategies. Participants communicated strategies they employed for designing grasp-proximate interfaces without the support of GraV. For instance, some grasped a real object that matched the object in the design task and simulated interacting with the UI while grasping that object to understand the interaction intricacies. P1 - *“I am not sure how precise users could be. I needed to try to hold the object myself because otherwise, I would not have known how to hold it.”* Others said they simply imagined how far a user’s finger would reach and used that as a strategy to help them place the UI widgets. Participants restated the assumption about interface element priority and described the approach of placing high-priority elements next to fingers assumed to be dexterous to support easy access to high-priority features, but other participants mentioned they decided to place assumed high-priority elements closer to fingers that had less assumed influence on the grasp. P3 - *“I tried to ensure the grip wouldn’t be impacted by clicking buttons.”*

6.2.2 With-GraV. Here we present the gathered information during the second discussion session and individual questionnaires based on the With-GraV design task. Figure 12 shows the changes in participant designs using GraV from designs without-GraV.

Updated Assumptions. Participants declared that the grasp interaction volume replaced their finger reach assumptions. P2 wrote: *“(GraV) helped counteract a tendency to overestimate the flexibility of user hand and finger movements.”* Participants voiced that after having access to GraV they rescaled the interface elements to fit within its boundaries. They stated that GraV provided data about which fingers were able to move by how much when grasping an object.

Revised Strategies. Some participants stated that they used GraV to refine the element positioning of their initial layout. P9 wrote, *“I could determine the button positions based on the visualized accessible area.”* They mentioned favoring low-cost regions of the volume as a general strategy. Some participants expressed they placed elements of assumed high importance in the low-cost regions of volume. P4 - *“Knowing the low and high-cost zones allowed me to place the buttons based on their importance/priority/frequency of being pressed.”* Others communicated they placed high-importance elements in the high-cost regions of the volume to avoid accidental interactions. *“High-cost zones are less prone to false positives and allow smaller buttons”* - P9.

Expressed Benefits. Participants pointed out that GraV helped them better understand hand types that were different from their own. When asked how GraV would change their design process, P1 answered: *“I would maybe try to use less of my own assumptions about body movements when for example designing for a child’s hand or a person with arthritis.”* Participants noted that they were able to make faster layout decisions with GraV. They also described how GraV helped them to identify design opportunities such as using fingers to interact in directions not originally considered such as moving outward and using the fingernails as the selected input part. They also said that it was helpful to consider other fingers, besides the thumb and index, for interaction. P1’s design presents an example of this in the *Interaction Change* row in Figure 12.

7 DISCUSSION

In this section, we expand on the outcomes of the Design Workshop (Section 5). Overall, the resulting opinions of the designers resonate with the design factors and the intended use cases for GraV. Designers expressed that GraV provided valuable information that could assist them in better understanding the hand-object interaction and depend less on assumptions.

7.1 Elementary Approaches to GPU design

Throughout the introductory design task, we noted that the assumptions and strategies employed by participants were based on their levels of confidence and prior experience with designs akin to XR interfaces for physical tasks while grasping an object.

Initially, designers (P2, P5, P6, P7) expressed a lack of confidence or familiarity with such interfaces. They tended to employ a familiar mid-air approach, while still ensuring they could maintain a grasp on the object. In contrast, participants (P1, P3, P4, P8, P9)

who articulated concerns regarding interface visibility during object usage or exhibited some degree of familiarity with the design scenario opted for a more hand-proximate design approach. This approach is exemplified in Figure 11 in the P4-G example. Furthermore, it is noteworthy that two designers with backgrounds in digital fabrication or manual crafting incorporated the object surface to provide haptic feedback when interacting with the UI widgets. These findings provide insights into the challenges experienced designers may encounter when attempting to diverge from established spatial design practices, inspired by 2D UI design [9], particularly when engaging with designs tailored for single-handed interactions. They also show how real-world familiarity with hand-object interactions could help designers feel more comfortable with GPU design.

We observed that the challenges expressed by the designers in both the introductory task (Phase 1) and the without-GraV design task (Phase 2) reflected the factors presented in Section 3. When we asked participants what information they used to complete their designs, they mentioned that knowing how the user could reach the interface was crucial. P8 said, *“I tried to understand which fingers have more flexibility to move and what is the range of movement available”* - P8 continued, *“I feel like having the haptic of the knife -oh I touched the knife... feels potentially important.”* P5 shared the following, *“I think the biggest challenge was thinking about how close or far the buttons should be from the hand since I had to balance between accidentally clicking the buttons and keeping them close enough to hit.”* This shows considerations that correlate with the Finger Motion Range and the Reachable Interaction Volume factors presented in Section 3, before participants were introduced to GraV. P1 expressed concerns about the lack of clarity on the possible movements and how holding a physical object was useful to get a more tangible idea, saying, *“Figuring out possible outcomes of movement, to me it was extremely helpful to hold the objects and I cannot imagine what it would be like to try to design without holding the actual object... it’s hard to think of a UI that would actually work.”* We believe that for cases like this, the boundary conditions and reachability values can offer a clearer perspective on the limitations for both the object and the reach of the fingers.

7.2 Towards GraV Adoption

Designers shared how they would use it in their personal design workflows and future applications including expressed interest in a model that could incorporate more information about the biomechanics of the hand, additional visual elements (e.g., animated visuals of the hand behavior reaching certain areas of the space), and even the level of visibility of certain UI elements.

Some of the designers highlighted that GraV inspired them to design interactions for the back of the fingertip (specifically the fingernail). This departure from commonly used interaction modalities for XR interfaces was particularly notable. Some designers integrated this strategy into their UI designs as presented by Figure 12 in the Goal Change row with pitcher design, where the designer places a button meant to be reached by the back of the thumb when using GraV.

Some designers highlighted the fact that the interaction costs visualized in the GraV had the potential to streamline certain parts

of the design process. P1 said, *“If I am going to see a point cloud I might as well just send that to an algorithm and ask it to optimize (...)”*, suggesting they foresee that GraVs could become input for some kind of UI layout optimization algorithm. Expanding on this possibility we envision GraV becoming a common exchange format, visually meaningful for humans and machine-readable, fostering human-AI design collaboration. We also believe this could facilitate collaboration across designer teams developing interfaces for various applications under the same hand-object coupled interaction, by providing a quantifiable common ground about their target user scenario. Designers commented on wanting to integrate this with other interaction modalities. P4 said, *“This combined with gestures could be a pretty powerful way to reduce the fatigue caused by multi-limb movement in AR and VR.”*

8 LIMITATIONS & FUTURE WORK

Currently, we simulate finger motions independently from each other, i.e., a finger does not interfere with other fingers during the simulation, they only collide with themselves, the palm, or the handheld object. Given this limitation, our simulation cannot generate GraVs of interactions that require coordinated motion of multiple fingers, e.g., finger-thumb gestures. To accomplish that, it would be necessary to revise the definition of a valid fingertip position to consider the joint orientations of other fingers. Additionally, the simulation does not generate GraVs that represent involuntary co-dependent movement. Fingers are not completely independent from each other and present mirror movements, which are simultaneous involuntary movements in response to the voluntary movement of a different finger [36]. This phenomenon has implications for GPU design because it can cause accidental interactions, known as the Midas Touch problem [23]. Future work can consider biomechanically accurate simulations using musculoskeletal models, such as OpenSim [7], to simulate more authentic hand-object interactions. This approach can inform designers of finger co-dependent movement and generate new types of finger motion costs.

Another limitation of our current method for generating GraV is that it does not simulate Physics and focuses only on collisions. There are several variables to be considered for a physically accurate simulation including weight distribution along the object, static friction between hand and object, static equilibrium, and skin and muscle deformation along with the held object parameters. The designers recognized that the reachability was crucial information that is not evident in contrast to visible object elements (e.g., dangerous areas, existing physical buttons, moving parts, etc.).

As presented in Section 6, designers found that GraV enhanced their awareness of the reachable space of a grasping hand. Although reachability is crucial for designing GPUIs, it is not the only factor. For example, keyboard designs often trade some reachability for better visibility. Future research should explore factors beyond those mentioned in this paper. A promising direction is the study of transitions between grasp types, which occur when users accommodate objects in their hands. Our design workshop considered two grasp types for the same hand-object pairing, but further studies on grasp transitions could enhance GPU design process.

9 CONCLUSION

In this paper, we introduce the motivation of how XR interfaces, for physical tasks, could benefit from adopting a similar design approach of tools and other everyday objects with integrated interfaces. Based on that we present the idea of Grasp-Proximate User Interfaces (GPUIS), a series of influencing factors, and a design space to assist designers in approaching these types of UIs.

We also present Grasp Interaction Volume (GraV) for UI designers, which represents the 3D space that can be reached by the hand or fingers while holding an object, presented as a point cloud that encodes the joint rotation cost at each point. To provide access to GraVs, we built a dataset of GraVs, based on anthropometric values and common everyday objects, and a finger motion simulation tool in Unity (GraVSim) to generate new GraVs. Results from a workshop design evaluation of GraV with XR designers indicate the benefits and value provided by the interaction volume data for supporting the design of GPUIS.

ACKNOWLEDGMENTS

This work was supported by the U.S. National Science Foundation (Early CAREER Award 2023 no. 2112633). We thank Ana Cardenas, You-Jim Kim, and Ashley Del Valle for their helpful feedback and discussions.

REFERENCES

- [1] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K Chilana. 2020. Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.
- [2] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. 161–168. <https://doi.org/10.1145/1738826.1738853>
- [3] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. 2015. The YCB object and Model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*. 510–517. <https://doi.org/10.1109/ICAR.2015.7251504>
- [4] E. Y Chao. 1989 - 1989. *Biomechanics of the hand : a basic research study*. World Scientific, Singapore ;.
- [5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9044–9053.
- [6] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégoire Rogez. 2020. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5031–5041.
- [7] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* 54, 11 (2007), 1940–1950.
- [8] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 289–300.
- [9] João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtnner, and Kaj Grønbæk. 2021. XRonomics: facilitating the creation of ergonomic 3D interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11. <https://doi.org/10.1145/3411764.3445349>
- [10] Shariff AM Faleel, Michael Gammon, Kevin Fan, Da-Yuan Huang, Wei Li, and Pourang Irani. 2021. Hpui: Hand proximate user interfaces for one-handed interactions on head mounted displays. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4215–4225.
- [11] Shariff AM Faleel, Michael Gammon, Yumiko Sakamoto, Carlo Menon, and Pourang Irani. 2020. User gesture elicitation of common smartphone tasks for hand proximate user interfaces. In *Proceedings of the 11th Augmented Human International Conference*. 1–8. <https://doi.org/10.1145/3396339.3396363>
- [12] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kauffmann, Michael J Black, and Otmar Hilliges. 2023. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12943–12954.
- [13] Cathy Mengying Fang and Chris Harrison. 2021. Retargeted self-haptics for increased immersion in vr without instrumentation. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 1109–1121. <https://doi.org/10.1145/3472749.3474810>
- [14] Thomas Feix, Javier Romero, Heinz-Bodo Schmidmayer, Aaron M Dollar, and Danica Kragic. 2015. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems* 46, 1 (2015), 66–77.
- [15] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 441–450.
- [16] Chris Harrison, Shilpa Ramamurthy, and Scott E Hudson. 2012. On-body interaction: armed and dangerous. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. 69–76. <https://doi.org/10.1145/2148131.2148148>
- [17] Rex Hartson and Partha S. Pyla. 2012. Chapter 14 - Rigorous Empirical Evaluation: Preparation. In *The UX Book*, Rex Hartson and Partha S. Pyla (Eds.). Morgan Kaufmann, Boston, 503–536. <https://doi.org/10.1016/B978-0-12-385241-0-00014-2>
- [18] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11807–11816.
- [19] Fengming He, Xiyun Hu, Jingyu Shi, Xun Qian, Tianyi Wang, and Karthik Ramani. 2023. Ubi Edge: Authoring Edge-Based Opportunistic Tangible User Interfaces in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [20] Steven J Henderson and Steven K Feiner. 2007. Augmented reality for maintenance and repair (armar). *Columbia Univ New York Dept of Computer Science* (2007).
- [21] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1957–1967.
- [22] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. Digitspace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1526–1537. <https://doi.org/10.1145/2858036.2858483>
- [23] Howell Istance, Richard Bates, Aulikki Hyrskykari, and Stephen Vickers. 2008. Snap clutch, moded approach to solving the Midas touch problem. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. 221–228.
- [24] Rahul Jain, Jingyu Shi, Runlin Duan, Zhengzhe Zhu, Xun Qian, and Karthik Ramani. 2023. Ubi-TOUCH: Ubiquitous Tangible Object Utilization through Consistent Hand-object interaction in Augmented Reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (, San Francisco, CA, USA.) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 12, 18 pages. <https://doi.org/10.1145/3586183.3606793>
- [25] Nikhita Joshi, Parastoo Abtahi, Raj Sodhi, Nitzan Bartov, Jackson Rushing, Christopher Collins, Daniel Vogel, and Michael Glueck. 2023. Transferable Micro-gestures Across Hand Posture and Location Constraints: Leveraging the Middle, Ring, and Pinky Fingers. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.
- [26] Woojoo Kim and Shuping Xiong. 2022. Pseudo-haptics and self-haptics for freehand mid-air text entry in VR. *Applied Ergonomics* 104 (2022), 103819. <https://doi.org/10.1016/j.apergo.2022.103819>
- [27] Panayiotis Koutsabasis and Panagiotis Vogiatzidakis. 2019. Empirical research in mid-air interaction: A systematic review. *International Journal of Human-Computer Interaction* 35, 18 (2019), 1747–1768.
- [28] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional.
- [29] Kyung-Sun Lee and Myung-Chul Jung. 2015. Ergonomic evaluation of biomechanical hand function. *Safety and health at work* 6, 1 (2015), 9–17.
- [30] Yuwei Li, Longwen Zhang, Zesong Qiu, Yingwenqi Jiang, Nianyi Li, Yuexin Ma, Yuyao Zhang, Lan Xu, and Jingyi Yu. 2022. NIMBLE: A Non-Rigid Hand Model with Bones and Muscles. *ACM Trans. Graph.* 41, 4, Article 120 (jul 2022), 16 pages. <https://doi.org/10.1145/3528223.3530079>
- [31] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and EW Looney. 2004. Twiddler typing: One-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 671–678. <https://doi.org/10.1145/985692.985777>
- [32] I. Scott MacKenzie. 2013. Chapter 5 - Designing HCI Experiments. In *Human-computer Interaction*, I. Scott MacKenzie (Ed.). Morgan Kaufmann, Boston, 157–189. <https://doi.org/10.1016/B978-0-12-405865-1.00005-4>

[33] Andrew T Miller and Peter K Allen. 2004. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine* 11, 4 (2004), 110–122.

[34] Pranav Mistry, Pattie Maes, and Liyan Chang. 2009. WUW-wear Ur world: a wearable gestural interface. In *CHI'09 extended abstracts on Human factors in computing systems*. 4111–4116. <https://doi.org/10.1145/1520340.1520626>

[35] Pedro Monteiro, Guilherme Gonçalves, Hugo Coelho, Miguel Melo, and Maximino Bessa. 2021. Hands-free interaction in immersive virtual reality: A systematic review. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2702–2713.

[36] Nilesh A Nadkarni and Shrikant S Deshmukh. 2012. Mirror movements. *Annals of Indian Academy of Neurology* 15, 1 (2012), 13.

[37] Riccardo Palmarini, John Ahmet Erkoyuncu, Rajkumar Roy, and Hosein Torab-mostaedi. 2018. A systematic review of augmented reality applications in maintenance. *Robotics and Computer-Integrated Manufacturing* 49 (2018), 215–228.

[38] Francisco Perella-Holfeld, Shariff AM Faleel, and Pourang Irani. 2023. Evaluating design guidelines for hand proximate user interfaces. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 1159–1173.

[39] Alexander Plopski, Teresa Hirzle, Nahal Norouzi, Long Qian, Gerd Bruder, and Tobias Langlotz. 2022. The eye in extended reality: A survey on gaze interaction and eye tracking in head-worn extended reality. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–39.

[40] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, and Karthik Ramani. 2022. ARnnotate: An Augmented Reality Interface for Collecting Custom Dataset of 3D Hand-Object Interaction Pose Estimation. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[41] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Trans. Graph.* 36, 6, Article 245 (nov 2017), 17 pages. <https://doi.org/10.1145/3130800.3130883>

[42] Adwait Sharma, Michael A Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. 2021. SoloFinger: Robust microgestures while grasping everyday objects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15. <https://doi.org/10.1145/3411764.3445197>

[43] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping microgestures: Eliciting single-hand microgestures for handheld objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13. <https://doi.org/10.1145/3290605.3300632>

[44] Alvy Ray Smith. 1979. Tint fill. In *Proceedings of the 6th annual conference on Computer graphics and interactive techniques*. 276–283.

[45] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. Fingerinput: Capturing expressive single-hand thumb-to-finger microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. 177–187. <https://doi.org/10.1145/3279778.3279799>

[46] David J Sturman and David Zeltzer. 1994. A survey of glove-based input. *IEEE Computer graphics and Applications* 14, 1 (1994), 30–39. <https://doi.org/10.1109/38.250916>

[47] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. 2020. GRAB: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV* 16. Springer, 581–600.

[48] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 581–600.

[49] Bruce Thomas, Karen Grimmer, Joanne Zucco, and Steve Milanese. 2002. Where does the mouse go? An investigation into the placement of a body-attached touchpad mouse for wearable computers. *Personal and Ubiquitous computing* 6 (2002), 97–112. <https://doi.org/10.1007/s007790200009>

[50] Matthieu B Trudeau, Tawan Udtamadilok, Amy K Karlson, and Jack T Dennerlein. 2012. Thumb motor performance varies by movement orientation, direction, and device size during single-handed mobile phone use. *Human factors* 54, 1 (2012), 52–59. <https://doi.org/10.1177/0018720811423660>

[51] Ying-Chao Tung, Chun-Yen Hsu, Han-Yu Wang, Silvia Chyou, Jhe-Wei Lin, Pei-Jung Wu, Andries Valstar, and Mike Y Chen. 2015. User-defined game input for smart glasses in public space. In *Proceedings of the 33rd Annual ACM conference on human factors in computing systems*. 3327–3336.

[52] Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. 2015. Intelligent augmented reality training for motherboard assembly. *International Journal of Artificial Intelligence in Education* 25 (2015), 157–172.

[53] Eric Whitmire, Mohit Jain, Divyee Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–21. <https://doi.org/10.1145/3130978>

[54] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–10. <https://doi.org/10.1145/3173574.3173752>

[55] Xuhai Xu, Alexandru Danu, Pattie Maes, and Suranga Nanayakkara. 2018. Hand range interface: Information always at hand with a body-centric mid-air input surface. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–12. <https://doi.org/10.1145/3229434.3229449>

[56] Goshiro Yamamoto and Kosuke Sato. 2007. PALMbit: a body interface utilizing light projection onto palms. *映像情報メディア学会* 61, 6 (2007), 797–804. <https://doi.org/10.3169/itej.61.797>

[57] Qian Zhou, Sarah Sykes, Sidney Fels, and Kenrick Kin. 2020. Gripmarks: Using hand grips to transform in-hand objects into mixed reality input. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11. <https://doi.org/10.1145/3313831.3376313>

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009