



**HAL**  
open science

# Alfred: the Haptic Butler On-Demand Tangibles for Object Manipulation in Virtual Reality using an ETHD

Victor Rodrigo Mercado, Thomas Howard, Hakim Si-Mohammed, Ferran Argelaguet Sanz, Anatole Lécuyer

► **To cite this version:**

Victor Rodrigo Mercado, Thomas Howard, Hakim Si-Mohammed, Ferran Argelaguet Sanz, Anatole Lécuyer. Alfred: the Haptic Butler On-Demand Tangibles for Object Manipulation in Virtual Reality using an ETHD. WHC 2021 IEEE World Haptics Conference, Jul 2021, Montreal, France. pp.373-378, 10.1109/WHC49131.2021.9517250 . hal-03469115

**HAL Id: hal-03469115**

**<https://inria.hal.science/hal-03469115>**

Submitted on 7 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Alfred: the Haptic Butler

## On-Demand Tangibles for Object Manipulation in Virtual Reality using an ETHD

Víctor Rodrigo Mercado<sup>1</sup>▲, Thomas Howard<sup>2</sup>▲, Hakim Si-Mohammed<sup>3</sup>, Ferran Argelaguet<sup>4</sup> and Anatole Lécuyer<sup>4</sup>

**Abstract**—We present “Alfred”, a novel haptic paradigm for object manipulation in 3D immersive virtual reality (VR). It uses a robotic manipulator to move tangible objects in its workspace such that they match the pose of virtual objects to be interacted with. Users can then naturally touch, grasp and manipulate a virtual object while feeling congruent and realistic haptic feedback from the tangible proxy. The tangible proxies can detach from the robot, allowing natural and unconstrained manipulation in the 3D virtual environment (VE). When a manipulated virtual object comes into contact with the virtual environment, the robotic manipulator acts as an encounter-type haptic display (ETHD), positioning itself so as to render reaction forces of the environment onto the manipulated physical object. Here, we discuss the concept for this novel approach and present a simplified prototype using a single detachable tangible proxy supported by a UR5 industrial robot. Through illustrative use-cases in VR and a preliminary performance evaluation, we discuss implications for robot control and design of interaction techniques. We show that Alfred is adaptable to a wide range of virtual environments and interaction scenarios, making it a promising approach for haptic-enabled manipulation in VR, although system latency is a limitation that still remains to be addressed.

### I. INTRODUCTION

A major challenge in immersive virtual reality (VR) is that of providing users with the “feel” of virtual objects being manipulated without compromising VR’s other key features, in particular the freedom it provides in the design and reconfiguration of virtual environments (VEs), a property described as *malleability* by Poupyrev et al. [1]. Arguably, one of the most ecological solutions to this challenge comes in the form of haptic devices which resist user’s movements, preventing the interpenetration of a user’s hand and virtual objects. These devices mimic reaction forces encountered in real-world exploration and manipulation of 3D objects and have been found to greatly enhance the perceived realism of VEs and user’s sense of presence (e.g. [2]). These devices fall into three broad categories: tangible elements, robots, and hybrids between them. First, approaches using tangible elements encompass what is referred to as passive haptics (e.g. [3]) as well as holdable devices (e.g. [4]). These approaches rely on superimposing virtual objects with physical objects mimicking their shape and optionally other physical properties. They can be simple and inexpensive but require complex calibration procedures and can quickly become unwieldy for representing complex virtual environments because of the required number of tangible elements. Furthermore, they tend to limit malleability as re-configuring the VE requires the reconfiguration of the associated tangible environment. However, several solutions have been explored to overcome these limitations and provide a compromise between technical feasibility and adaptability to different VEs, including the use of mismatching tangible elements (e.g. [5]), retargeting techniques (e.g. [6], [7]), built-in actuation (e.g. [8], [9]) and external actuation (e.g. [10], [11]), as well as the use

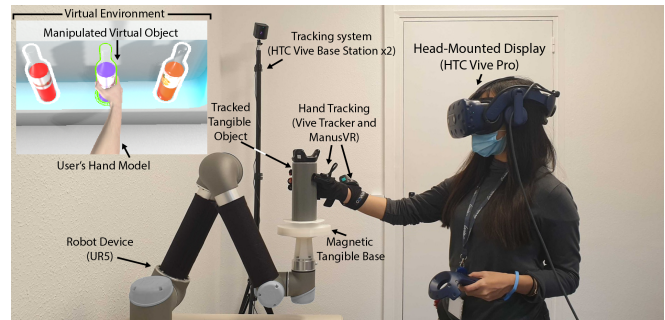


Fig. 1. The Alfred prototype: A user manipulates virtual objects by grasping and removing a tangible from the robotic tray. The Vive tracker mounted on the tangible allows precise reproduction of the tangible’s motions on the manipulated virtual object. Users wear an HTC Vive HMD displaying the virtual environment as shown in the upper left corner. A Vive tracker is strapped to the dorsal face of their hand equipped with a ManusVR tracking glove to precisely track and animate their virtual hand.

of elements already present in the physical space (e.g. [12]). Secondly, robotic approaches forego the reproduction of virtual objects in a physical form in favor of selectively applying forces to parts of a user’s body using robots, mimicking reaction forces encountered in real-world exploration and manipulation of 3D objects. These approaches usually provide a high degree of malleability at the cost of increased system complexity. Also, because of the actuation and control’s complexity, robotic approaches constrain interaction by limiting it to a small set of possible contact points between a user’s hand model and virtual objects. Robotic approaches can maintain permanent contact between their end-effectors and the user, as is the case for traditional force-feedback devices (e.g. [13], [14]), or can come into contact with the user as needed in the case of encounter-type haptic devices (ETHDs, e.g. [15], [16]). Furthermore, these systems may be grounded (e.g. [17], [18]), held (e.g. [19]), or worn by the user (e.g. [20], [21]).

Finally, a recent development has been the use of robotic manipulators with manipulable tangible elements as end-effectors (e.g. [22], [23], [24]), yielding hybrid approaches between tangibles and robotics. These interfaces overcome many limitations of robotic approaches when it comes to faithfully rendering grasping of virtual objects, allow intermittent contact with objects and maintain the repositioning freedom and malleability of robotic approaches.

The work presented here, builds on this last category of hybrid approaches. This paper presents Alfred – “the haptic butler for VR”: an industrial collaborative robot holding passive tangible objects that attach and detach from the robotic tray as needed (See Fig. 1), providing highly adaptable and faithful haptic rendering for manipulation and exploration of 3D objects in VR.

### II. CONCEPT

Without haptics, object manipulation in VR feels hollow and unreal, compromising user immersion and their feeling of presence in the VE. Alfred combines tangible objects and ETHDs in order to give users the illusion of a fully tangible virtual environment without compromising the VE’s malleability. It provides a high degree of freedom for VE reconfiguration while keeping the system’s complexity (hard- and software)

This project has been funded by the Agence Nationale de la Recherche (ANR) LobbyBot project and the Tactility project funded from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 856718. ▲Both authors contributed equally to this work.

<sup>1</sup> is with Univ Rennes, INSA, Inria, CNRS, IRISA – Rennes, France.

<sup>2</sup> is with Univ Rennes, CNRS, Inria, IRISA – Rennes, France.

<sup>3</sup> is with Univ. Lille, CNRS, Centrale Lille, CRISTAL – Lille, France

<sup>4</sup> are with Univ Rennes, Inria, CNRS, IRISA – Rennes, France.

E-mail: {victor.mercado, thomas.howard, ferran.argelaguet, anatole.lecuyer}@inria.fr

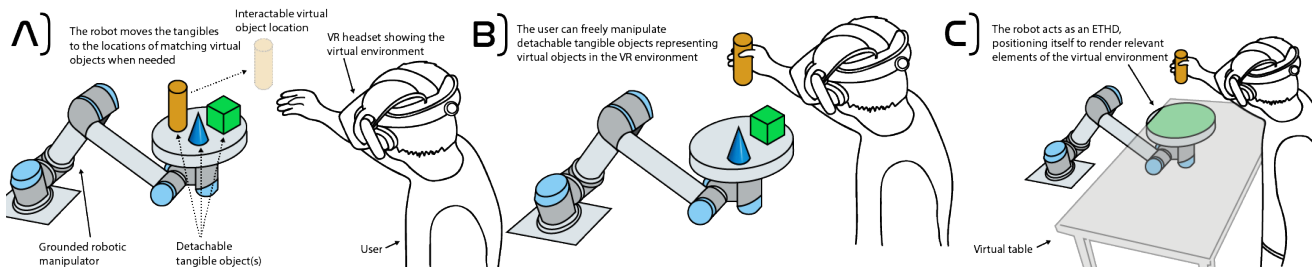


Fig. 2. Concept illustration for Alfred: (A) A robotic manipulator holds a set of tangible proxies capable of matching the properties of interactable virtual objects, placing them as required so they match the pose of virtual objects the users interacts with. (B) Upon grasping, a tangible proxy detaches from the robot tray and becomes freely manipulable. (C) During manipulation, the robot tray acts as an ETHD, positioning itself to render portions of the VE, e.g. allowing the user to put down the manipulated object.

manageable. The concept behind Alfred is that a set of detachable tangible proxies are held on a robotic manipulator which is capable of displacing and reorienting them in 3D to match the positions and orientations of corresponding virtual objects in an immersive 3D environment (see Fig. 2-A). Users begin by moving their hand in free space, while the system tries to anticipate the possible physical interactions by placing tangible elements in the required locations. Since the tangible proxies are not permanently fixed to the robotic tray, users can freely manipulate them by picking them up (see Fig. 2-B). Haptic feedback during manipulation is provided by the physical prop, in an approach akin to passive haptics, while the robotic tray continues to act as an ETHD. During the manipulation phase, the robot anticipates further possible physical interactions between the manipulated object and the remainder of the VE. When the manipulated virtual object comes into contact with other parts of the virtual environment, the robotic tray is positioned such as to render reaction forces of the virtual environment onto the manipulated virtual object (see Fig. 2-C). This allows users to e.g. place manipulated virtual objects on virtual surfaces such as tables as if they were truly present in the user’s physical environment. Key features provided by Alfred are: (1) infinite re-spawn of virtual objects, (2) removal of virtual objects, (3) multiple virtual object mapping using only one physical proxy as long as their geometries are similar enough, and (4) free reconfiguration of the VE. These features are illustrated in the implementation of Alfred presented in Section III. This concept takes McNeely’s idea of “robotic shape displays” (industrial robots equipped with a cache of actual objects on their end-effectors [25]) one step further, making these objects detachable to allow free manipulation without the complexity of dealing with making the robotic manipulator transparent. Contrary to grounded [22], [24] or wearable [23] ETHDs with tangible objects as end-effectors, our approach does not constrain manipulation to a limited subset of object grasp poses, allowing a wide range of natural manipulations of an object once grasped. Also, Alfred decouples the actuation from the tangible object, setting it apart from mobile robotic proxies [8], [9] which incorporate actuation inside tangible objects, as well as robotic manipulators with fixed tangible end-effectors [24]. In our contribution the tangible only acts as a physical proxy for the virtual object, and the robotic manipulator repositions these proxies in the 3D environment. It therefore becomes possible to pick up and put down tangibles anywhere in the 3D environment, without any additional constraints regarding physically pairing the VE and the underlying physical environment. The only additional element occupying physical space is the robotic manipulator, which has the advantage of being mobile, allowing it to get out of the user’s way when necessary.

### A. System components

Alfred builds on three sets of components : Physical components, components ensuring co-location between physical and virtual elements, and interaction components.

The physical components are responsible for providing the tangible representation of the VE. They encompass the robotic manipulator, which

may be a robotic arm (as shown in Fig. 2), the set of tangible objects and a system for attaching and carrying the tangibles on the robot’s end-effector.

The robotic manipulator must be capable of moving the set of tangible objects to poses matching those of virtual objects. Consequently, its workspace should overlap with the region of the VE in which interaction and manipulation take place. Furthermore, its end-effector should have sufficient degrees of freedom (DoFs) to match all possible positions and orientations of interactable virtual elements with those of the physical proxies. Finally, its actuation should allow it to carry the payload of physical proxies while ensuring sufficiently high accelerations and velocities to ensure the tangibles can be brought to target poses in a timely fashion.

The system for attaching the tangible objects to the robot’s end effector should accommodate the required number of tangibles while ensuring that they are presented individually, avoiding unintended contact between the hand and other physical elements during interaction. It should also secure the tangibles in place on the end-effector during relocation, while still allowing them to seamlessly detach upon grasping by the user.

Finally, the tangible objects must sufficiently match the shapes (and possibly other physical properties) of the virtual objects to be represented, or should implement solutions for compensating mismatches (such as e.g. use of visual-haptic perceptual illusions [5], actuation for shape changing [26], augmentation with other haptic actuators [11]).

The second set of components is responsible for establishing and maintaining co-location between physical and virtual elements, while avoiding discontinuities in virtual elements’ poses. It encompasses registration, tracking and motion planning components.

Registration components ensure an initial match between the VE and physical environment. This includes calibration of the user’s hand model, of interactable virtual objects with respect to the tangibles, as well as of the robot’s pose and motion with respect to the VE.

The tracking components include sensors and software for keeping track of the poses of the user and all physical elements to appropriately render their virtual counterparts. The tracking components’ outputs also enable motion planning and provide input for the interaction components. To maintain coherence between user motor input and perceived visual and haptic feedback, tracking accuracy, low latency and high update rate are essential requirements. Furthermore, the software fusing input from multiple sensors must ensure continuity and coherence of the rendered virtual environment.

Motion planning software components are responsible for generating robot motion commands ensuring the tangibles are brought to the correct poses as precisely and as fast as possible. These commands must also ensure a match between the properties of the virtual objects’ motions and the tangible objects’ motions. Finally, these components are responsible for preventing unwanted contact between the user and the robot or tangibles.

The third set of components, i.e. interaction components, encompass the three components of a manipulation interaction technique (IT) as defined by Bowman et al. [27] (selection, manipulation and release) as well as a

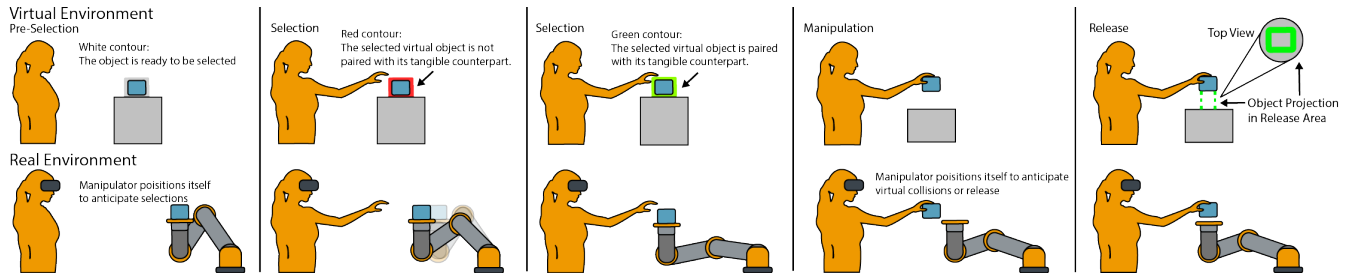


Fig. 3. Our implementation of Alfred’s interaction technique. Pre-selection: Selectable virtual objects show a white contour, the robot positions its end-effector to best anticipate future selections (see Fig.4). Selection: Objects are selected based on the distance of the hand to the virtual object. The selected object’s contour is red while Alfred has not aligned the tangible object with its virtual counterpart, and become green as soon as it has. Manipulation: The user freely manipulates the tangible object while the robot positions itself to anticipate collisions between object and environment. Release: Alfred’s end-effector encounters the tangible at a chosen release position.

pre-interaction component. The pre-interaction component fulfils the dual objective of making the user aware of interactable objects in the VE and of anticipating possible future selections, putting the system into an optimal state for responding to these selections. The selection component allows users to indicate to the system which virtual object they intend to interact with. This indication can be implicit based on constraints in the interaction design and assumptions made on the basis of user input, or may be explicitly provided through user input. The selection component also provides feedback indicating the feasibility and the success of a selection, e.g. through visual highlighting as discussed in Sec. III-C. The manipulation component is based on ecological manipulation of a tangible object. Thus, it is restricted to the object being attached to the user’s virtual hand and to the maintaining of the relation between the tracked tangible’s pose and that of the virtual object. Faithful haptic feedback is implicitly provided through the use of tangible proxies. The release component allows users to indicate to the system onto which part of the VE the manipulated object will be released. This can be anticipated because of constraints to the interaction (e.g. as discussed in Sec. III-C), can be inferred from user’s actions (see e.g. Sec. III-D) or can be explicitly pointed out (see e.g. Sec. III-B). This component provides feedback similarly to the selection component, indicating the feasibility and success of a release.

### B. Safety

Safety is paramount in applications involving close collaboration between robotic arms and humans, especially in VR applications where the user is blind to the actual position and motion of the robotic arm. Safety in close human-robot collaboration has been extensively studied (e.g. [28]) and has been standardized for robotic manufacturing in [29].

Implementations of Alfred should consider these requirements and ensure multiple levels of safety in their physical, co-location and interaction components to prevent any potential harm to the user such as e.g.: (1) Use of a robotic manipulator with inherent compliance and passive safety through low weights, inertia and speeds; (2) Reliable high frequency and minimal latency tracking and associated collision avoidance algorithms; (3) Dead-man switches, emergency stops and external supervision; (4) Interaction techniques warning users of potentially dangerous situations.

## III. ILLUSTRATIVE SETUP

To illustrate the concept described in Sec. II, we designed a simplified prototype of Alfred integrating the three sets of system components and illustrating its key features through several illustrative manipulation scenarios. We chose to restrict the scope of interactions to (1) the use of a single tangible prop and (2) virtual objects in co-planar arrangements.

### A. System prototype

In terms of physical components, the set-up (shown in Fig. 1) used a UR5 robot equipped with a 3D-printed end-effector acting as a tray for a single cylindrical tangible object. The tangible is held in place

using passive magnets embedded within its base and the tray, similarly to methods used in [5] and [3]. To ensure co-location, the tangible object is tracked using a Vive tracker, ensuring accurate reflection of manipulation movements onto the corresponding virtual object. The VE is displayed through an HTC Vive HMD, and their right hand is tracked using a Vive tracker and a ManusVR glove, allowing precise animation of their virtual hand. As previously mentioned in Sec. II-A, this Alfred prototype proposes an IT based on three interaction phases in addition to the pre-interaction phase (see Fig. 3), during which interactable objects are highlighted with a glowing white contour when the hand is close to them:

a) **Selection:** Object selection can be either constrained, in the case where there is only one interactable virtual object in the VE (see Sec III-B), or unconstrained if multiple virtual objects can be selected (see Secs. III-C and III-D). It can also occur explicitly (see Sec III-B) or implicitly e.g. by considering the distance between the virtual hand and closest virtual object (see Secs. III-C and III-D). If the user moves to grasp or touch a selected object before it has been brought into position by the robot, the contour becomes red, otherwise it becomes green as soon as the tangible is in position, signaling to the user that physical interaction is now possible (see Fig. 3).

b) **Manipulation:** Users can then freely touch, grasp and pick up the virtual object. In the latter case, the virtual model of the manipulated object is moved in the VE based on information from the tracker mounted on the tangible prop (see Fig. 1).

c) **Release:** Users explicitly (see Sec. III-B) or implicitly (see Secs. III-C and III-D) select the place where they can release the object. A color-coded circular highlight indicates the release position and the feasibility of a release by switching from red (see Fig. 5) to green (see Fig. III-D).

In this implementation, trajectory planning for the UR5 is kept simple: At every frame, the robot end-effector’s desired position  $RC$  is computed within a restricted interaction zone (see e.g. Fig. 4) based on the interaction phase, the user’s hand position and the position of interactable virtual objects. Once  $RC$  is known, the end-effector trajectory is computed to achieve a linear motion of the robot’s end-effector towards  $RC$ , using the Universal Robots’ SDK.

The illustrative scenarios presented in the following explore the design space for selection, release and mapping of tangible to virtual objects. They are depicted in the video provided as supplemental material.<sup>1</sup>

### B. Illustration 1: Manipulating one object between known positions

This first scenario is a virtual parts inspection task on an assembly line described in Fig. 5. It showcases the possibility for infinitely spawning virtual objects which will always have an existing physical counterpart, as well as the possibility for removing virtual objects from the environment without leaving problematic physical objects in the supposedly empty

<sup>1</sup><https://youtu.be/-cioXiJWBwU>



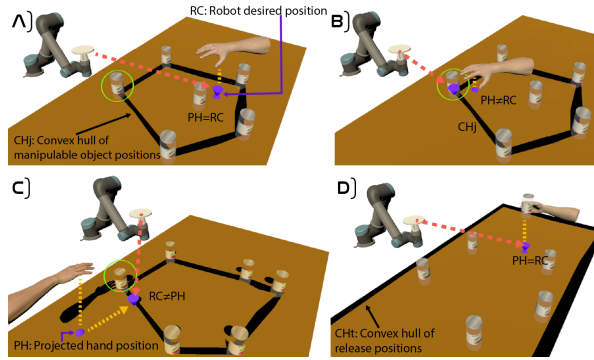


Fig. 4. Computation of the robot end-effector desired position (RC) when grasping and release positions are not known beforehand: When the user has no object in hand (cases A, B & C), the robot motion space is the inside of the convex hull  $CH_j$  formed by the object center positions in the plane supporting the objects, a table in this example. The user's hand position is projected in this plane (PH), if PH is within CJ (cases A & B), PH is used as the robot end-effector's position command (case A) unless the hand is close enough to an object (case B) in which case the robot desired position becomes that of said object. If PH lies outside CJ (case C), RC is computed as the point of CJ closest to PH. Similarly, once the user has an object in hand (case D), the robot motion space is delineated by  $CH_t$ , formed here by the four corners of the virtual table. RC is then computed based on PH, identically to cases A or C, depending on whether the hand is directly above the table or not. The release position is freely chosen within  $CH_t$  by the user, and the convex hull  $CJ$  is recomputed upon release of the object.



Fig. 5. Users face a table onto which a virtual part is extruded. They can pick it up and inspect it, then they select whether the part is flawed or good. This explicitly indicates their future to place the part either into the disposal bin or onto the conveyor belt. The robot drives to the chosen release position to preempt the user's release of the part, after which it removes the tangible object from the interaction space as the virtual part is either destroyed or removed from the scene.

virtual interaction space. This scenario constrains interactions by providing a single initial object position, and two possible fixed object destinations (the disposal bin and the conveyor belt for the good parts). Combined with the user's explicit indication of a future release position, this scenario uses a control scheme in which the robot carrying the tangible is initially driven to the location where the virtual part is extruded (i.e. initial position), then drives to the release position indicated by the user. Finally, the robot removes the tangible from the interaction space once the virtual part is either destroyed or removed. This scenario represents the most favorable configuration for the robot's control, as only one tangible object is used at a time and the robot end-effector's target location is known beforehand. However, this comes at the expense of a limited range of possible interactions, and cannot adequately respond to the user performing unpredictable releases.

### C. Illustration 2: Mapping one tangible to many virtual objects

This scenario showcases how Alfred allows mapping of a single tangible object to multiple simultaneously present virtual counterparts by letting a user manipulate one of many different virtual bottles in a bartending scenario (see Fig. 6). To allow this, we introduce additions to both the robot control and interaction technique. Regarding the control of the robot, since the tangible object can be mapped to virtual objects at different locations, the robotic arm has to anticipate the users' actions in order to ensure that the tangible object is placed at the desired location. In this sense, the robot behaves like a traditional ETHD. The user's hand

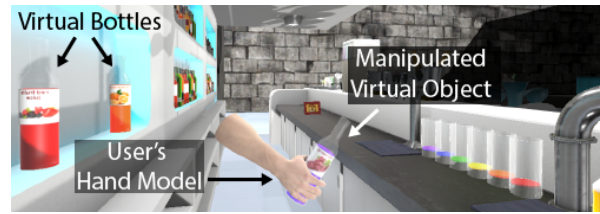


Fig. 6. Users embody a virtual bartender mixing virtual cocktails. They select one of six bottles arranged on the bar's shelf, which they manipulate to pour drinks, before placing them back at their original position.

position is projected onto the line passing through the centers of the virtual bottle bases, indicating the desired robot end-effector position. The end-effector follows the user's hand motion while remaining constrained to this line segment in the plane on which all virtual bottles are placed. This behavior ensures the robot always has the shortest distance to travel to the position of any bottle the user chooses to interact with. However, this robot control approach by itself runs the risk of introducing unwanted motions to the prop and positional mismatches if the user goes to rapidly grasp a bottle. The interaction technique is therefore improved by defining a selection region defined around each virtual bottle. If the user's hand comes within this region, it is assumed the user will grasp the associated bottle (selection) and the robot is immediately driven to the location where the tangible prop matches the bottle's pose. Furthermore, users always return bottles they pick up to the same location. The release position is implicitly defined at the moment of grasping, and the robot does not need to follow the user's motion during the free manipulation phase. It simply remains in position until the user puts the virtual bottle back. However, this approach where the release position is known in advance could easily be extended to releases at a different position.

### D. Illustration 3: Freely manipulating one of many objects

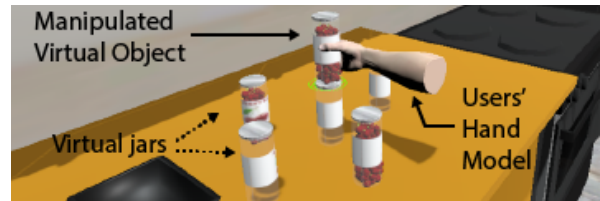


Fig. 7. Unconstrained manipulation of multiple objects. In this scenario the user faces a table holding five jars in arbitrary locations. Users are free to pick up any jar, manipulate it and place it back wherever they please, which includes stacking jars on top of each other.

This final scenario explores the use of Alfred for unconstrained manipulation of one of many virtual objects (see Fig. 7). Similarly to the previous scenario, a single tangible prop is mapped to multiple manipulable virtual objects. However, in this case the robot has to act as an ETHD during the free interaction phase, representing any surface or object in the environment that the manipulated object may come into contact with. Since only very few assumptions can be made about the user's intent and the interaction is unconstrained, this scenario showcases the most complex variant of control algorithm for the robot end-effector's desired position. If the robot followed the user's hand anywhere in the workspace, this could potentially bring the robot very far from the virtual objects to be represented, unnecessarily delaying subsequent interactions. Fig. 4 details the approach for calculating the robot end-effector's desired position in order to avoid this. In the event where a user wishes to grab a jar in a stack of jars, the height of the user's hand above the table is used to define whether the robot desired position is within the table plane (user wishes to grab the lower jar in the stack) or above the table plane (user wishes to grab one of the top jars in the stack).

#### IV. EVALUATION

The main criterion for the quality of haptic rendering using our system is the ability for the system to adequately anticipate contact between a user and a virtual object. Correct timing is essential to ensure that the tangible prop and end-effector are in position before the user’s hand comes into contact with the virtual element to be represented. The capability for an implementation of Alfred to comply with this timing constraint depends on numerous factors, including the robot’s dynamics, the size of the workspace, the VE configuration and complexity, the robot control scheme and IT (see Sec. III) as well as the user’s movement speeds.

##### A. Materials and methods

Considering Covid-19 restrictions, we performed an evaluation of the system’s rendering latency based on simulated robot motions computed from separately recorded user hand movements. Five participants (4 male, 1 female, ages 24-33, all right-handed) recorded planar radial movements when moving from a fixed start position  $P_{start}$  towards end positions  $P_{target}$ , and back (see Fig. 8), performing 3 repetitions of each motion. The users’ hands were tracked using an HTC Vive tracker providing position and orientation at a frame-rate of 90Hz. Recorded motions were then fed into a simulation of the robot (URSim running at a frame-rate of 125Hz), providing reference values for computing robot motions. Four different schemes for the end-effector’s position control were considered:

- Snapping (SP): the robot’s end-effector is driven to  $P_{target}$ , independently from the user’s hand position (as is the case in our first scenario discussed in Sec. III-B).
- Following (FW): the projection of the hand position onto the axis connecting  $P_{start}$  and  $P_{target}$  is set as the end-effector’s desired position. This approach is the simplest implementation of ETHD behavior.
- Snap After Threshold (ST): the robot is driven to  $P_{target}$  after the user’s hand crosses a threshold defined as the midpoint between  $P_{start}$  and  $P_{target}$ . This approach showcases behavior as it would occur in the case of distance-based selection of virtual objects.
- Snap After Follow (SF): this control scheme is a hybrid between the two aforementioned approaches. The end-effector follows the projection of the user’s hand. Once the hand crosses the threshold (midpoint between  $P_{start}$  and  $P_{target}$ ), the end-effector drives directly to  $P_{target}$ , independently from the user’s hand position. This control scheme is implemented in both scenarios described in Secs. III-C and III-D.

The simulator implemented realistic robot dynamics, considering an end-effector payload of 0.5kg, and the robot’s controller tuned to act as a critically damped system. We obtained 4 robot motion recordings and 1 hand motion recording for each trial, from which we computed the hand and robot’s respective arrival times at the target. Data for each distance and orientation pair were averaged over the 3 trial blocks for each user, yielding one point estimate per condition. Since no significant influence of orientation on hand arrival times was observed (pairwise Student’s t-tests do not reject  $H_0$  at  $\alpha = 0.05$ ), we considered trials at different orientations as repeats of trials at a given distance and computed the mean hand and robot arrival times as a function of target distance. The results are shown in Fig. 9. The recorded user motions are intended to provide orders of magnitudes of Alfred’s performance and are in no way meant to be exhaustive or fully representative.

##### B. Results

We obtained task completion times (i.e. times for the hand or robot to reach a target at a given distance) for the hand and the robot under each of the four considered control schemes (see Fig. 9) from which we inferred delay times for each control scheme, i.e. the difference in arrival times between the hand and robot in a given control scheme. The overall

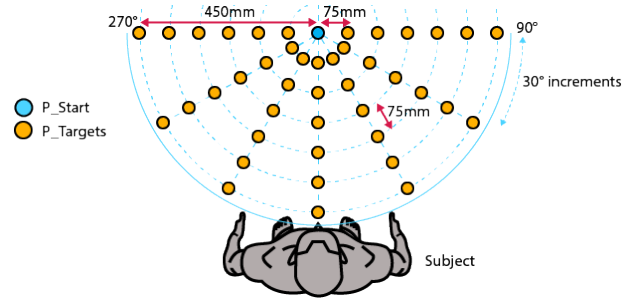


Fig. 8. Recorded user motions covered 6 distances between  $P_{start}$  (blue) and  $P_{target}$  (yellow) in 75mm increments over a range of 450mm (i.e. the full radius of the UR5’s workspace) for 7 directions, in 30° increments over a range of 180°.

average delay times for all conditions ranged approximately between 2.1s and 3.6s. The average task completion time for users varied between 0.42s and 1.24s, with mean performances across subjects as follows: 75mm  $M = 0.63s$ , 150mm  $M = 0.8s$ , 225mm  $M = 0.78s$ , 300mm  $M = 0.8s$ , 375mm  $M = 0.93s$ , 450mm  $M = 1.04s$ .

The results show that the SP position control provides the most reactive robot control for all distances (75mm  $M = 2.87s$ , 150mm  $M = 3.37s$ , 225mm  $M = 3.69s$ , 300mm  $M = 3.89s$ , 375mm  $M = 3.88s$ , 450mm  $M = 3.18s$ ). Pairwise Student t-tests between condition within distances show the observed differences are significant at  $\alpha = 0.05$ , except for the difference between SP and FW at 150mm. The longest times to arrive to the target were obtained when using the ST control scheme (75mm  $M = 3.5s$ , 150mm  $M = 4.07s$ , 225mm  $M = 4.27s$ , 300mm  $M = 4.41s$ , 375mm  $M = 4.47s$ , 450mm  $M = 3.84s$ ). Student t-tests between condition within distances show observed differences are significant at  $\alpha = 0.05$ , except for the difference between ST and FW at 450mm. The FW control scheme (75mm  $M = 3.11s$ , 150mm  $M = 3.63s$ , 225mm  $M = 3.83s$ , 300mm  $M = 4.01s$ , 375mm  $M = 4.$ , 450mm  $M = 3.76s$ ) and SF control scheme (75mm  $M = 3.06s$ , 150mm  $M = 3.59s$ , 225mm  $M = 3.83s$ , 300mm  $M = 4.01s$ , 375mm  $M = 4.02s$ , 450mm  $M = 3.57s$ ) yielded similar results for all distances.

For all control schemes, the difference in arrival times between hand and robot (i.e. the delay) increased with the length of motion, up to 375mm. Above this distance, the delay begins to drop, with the largest motions (450 mm) yielding the smallest measured delay for all the control schemes except FW (differences in delays between distances, within control schemes are significant at  $\alpha = 0.05$  except for FW - Pairwise Student’s t-tests).

#### V. DISCUSSION AND CONCLUSION

In this paper we propose Alfred, a hybrid approach which exploits the benefits of passive haptics and ETHDs. By remapping tangible objects to virtual objects, Alfred provides key features such as: re-spawn and removal of virtual objects, multiple virtual object mapping to a single tangible, and tangible elements’ reconfiguration in an immersive environment in VR. We implemented a prototype and 3 illustrative scenarios that portray how Alfred successfully provides haptic feedback in malleable VEs. To evaluate this approach’s performances, we simulated Alfred’s behavior for four representative robot control schemes. The results indicate that a snapping control scheme performs best on average. These results were expected due to the fact that the robot is prompted to go directly to the target rather than responding to a user’s motions. Following (FW) and threshold-based (ST, SF) approaches performed similarly well. The reason for SF performing worse than SP might be related to the robot’s positional error being relatively small at the beginning of the movement for snap after following when compared to snapping alone, which would mean lower robot speeds during motion. We considered the integration of real time hand following in the control schemes to see if the robot’s performance was faster under this condition. Snap After Threshold (ST) performed worst, which was expected since the robot has to wait until the hand position crosses a threshold to begin moving to the target position. In

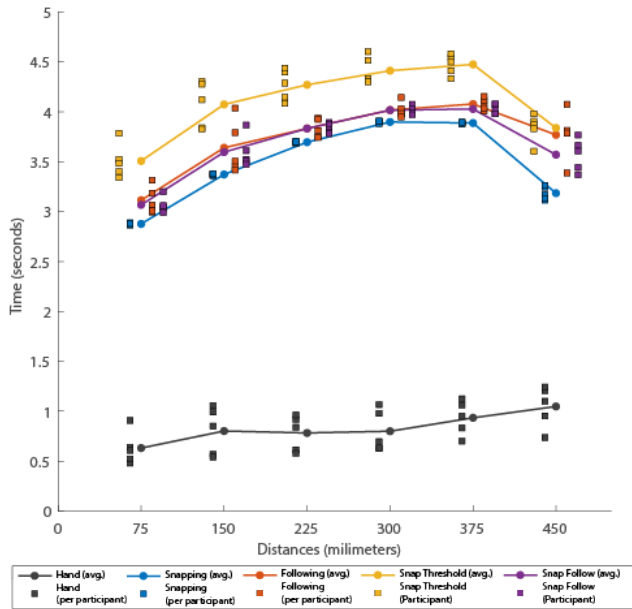


Fig. 9. Average time to reach the target position for the users' hand and robot end-effector under the 4 control schemes considered. The mean times for each motion distance are shown with a square symbol. The overall average for the hand motions and robot motions under each control scheme is plotted as a line showing the arrival time as a function of distance to the target.

real life scenarios in which the users' intent cannot be predicted, following approaches appear as the best choice considering the small differences with respect to the performances obtained with snapping. The advantages of the detachable tangible approach presents the challenge of properly tracking the object without compromising manipulation. In our current implementation, the tracker attached to the top of the tangible may hinder some object manipulations, thus future work will include improving tracking hardware to make it unobtrusive (e.g. using ArUco markers [30]). Latency is still a challenge to be addressed by the ETHD research community [31]. The delay for arriving at the target is a compound measure of human behavior and associated robot behavior and can thus be addressed through improvements to the robot hardware and control as well as through developed interaction techniques. Alfred could benefit from the integration of complementary approaches from the state of the art such as haptic-retargeting [6] to address speed and latency issues. Increasing the robot velocity would also require taking into account more stringent safety measures to avoid unwanted collisions with the user. For the moment our approach addresses these issues through constraints to the robot's motion space and the designed interaction technique which informs users' about the availability of tangible objects for manipulation. In parallel, it will be of interest to experimentally evaluate the impact of system delays for different interaction techniques on user experience and immersion. Future work could deal with improving our prototype hardware and interaction technique to investigate the following three points: (1) Increasing malleability by rendering manipulation of multiple different shaped objects using a wider assortment of detachable haptic proxies. (2) Researching implications of improved path planning and robot control methods that would allow rendering of haptic interaction with full 6DoF (position and orientation) in 3D space, while reducing system delay. (3) Enriching user interaction by allowing simultaneous manipulation of multiple objects, which could allow bi-manual and multi-user interaction with Alfred.

#### REFERENCES

- [1] I. Poupyrev *et al.*, "Actuation and tangible user interfaces: the vaucanson duck, robots, and shape displays," in *Proc. Int. Conf. Tangible & Embedded Interaction*, 2007, pp. 205–212.
- [2] B. E. Insko *et al.*, "Passive haptics significantly enhances virtual environments," Ph.D. dissertation, 2001.

- [3] H. Brument *et al.*, "Pyramid escape: Design of novel passive haptics interactions for an immersive and modular scenario," in *Proc. IEEE Conf. Virtual Reality & 3D User Interfaces (VR)*, 2019, pp. 1409–1410.
- [4] A. Zenner *et al.*, "Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality," *IEEE Trans. Visualization & Computer Graphics*, vol. 23, no. 4, pp. 1285–1294, 2017.
- [5] X. de Tinguy *et al.*, "How different tangible and virtual objects can be while still feeling the same?" in *Proc. IEEE World Haptics Conf.*, 2019, pp. 580–585.
- [6] M. Azmandian *et al.*, "Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences," in *Proc. Chi conf. on human factors in comp. sys.*, 2016, pp. 1968–1979.
- [7] L. Kohli, "Redirected touching: Warping space to remap passive haptics," in *Proc. IEEE Symp. on 3D User Interfaces*, 2010, pp. 129–130.
- [8] E. W. Pedersen *et al.*, "Tangible bots: interaction with active tangibles in tabletop interfaces," in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2011, pp. 2975–2984.
- [9] Z. He *et al.*, "Robotic haptic proxies for collaborative virtual reality," *arXiv preprint, arXiv:1701.08879*, 2017.
- [10] X. de Tinguy *et al.*, "Enhancing the stiffness perception of tangible objects in mixed reality using wearable haptics," in *Proc. IEEE Conf. Virtual Reality & 3D User Interfaces (VR)*, 2018, pp. 81–90.
- [11] S. V. Salazar *et al.*, "Altering the stiffness, friction, and shape perception of tangible objects in virtual reality using wearable haptics," *IEEE Trans. Haptics*, 2020.
- [12] A. L. Simeone *et al.*, "Substitutional reality: Using the physical environment to design virtual reality experiences," in *Proc. 33rd Ann. ACM Conf. Human Factors in Computing Systems*, 2015, pp. 3307–3316.
- [13] T. H. Massie *et al.*, "The phantom haptic interface: A device for probing virtual objects," in *Proc. ASME Symp. Haptic Interfaces for Virtual Environment & Teleoperator Systems*, vol. 55, no. 1. Chicago, IL, 1994, pp. 295–300.
- [14] G. Burdea *et al.*, "A portable dextrous master with force feedback," *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 1, pp. 18–28, 1992.
- [15] Y. Yokokohji *et al.*, "Designing an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behaviors," *Int. J. Robotics Research*, vol. 24, no. 9, pp. 717–729, 2005.
- [16] S. Nakagawara *et al.*, "An encounter-type multi-fingered master hand using circuitous joints," in *Proc. IEEE ICRA*, 2005, pp. 2667–2672.
- [17] M. Sato, "Spidar and virtual reality," in *Proc. 5th Biann. World Automation Congress*, vol. 13. IEEE, 2002, pp. 17–23.
- [18] E. Vonach *et al.*, "Vrrobot: Robot actuated props in an infinite virtual environment," in *Proc. IEEE Virtual Reality*, 2017, pp. 74–83.
- [19] E. o. Whitmire, "Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller," in *Proc. CHI Conf. Human Factors in Computing Systems*, 2018, pp. 1–12.
- [20] J.-B. Chossat *et al.*, "Soft wearable skin-stretch device for haptic feedback using twisted and coiled polymer actuators," *IEEE Trans. Haptics*, vol. 12, no. 4, pp. 521–532, 2019.
- [21] N. Nishimura *et al.*, "Wearable encounter-type haptic device with 2-dof motion and vibration for presentation of friction," in *Proc. IEEE Haptics Symposium*, 2014, pp. 303–306.
- [22] E. Ruffaldi, "Haptic rendering of juggling with encountered type interfaces," *Presence*, vol. 20, no. 5, pp. 480–501, 2011.
- [23] X. de Tinguy *et al.*, "Weatavix: Wearable actuated tangibles for virtual reality experiences," in *Proc. Eurohaptics Conf.*, 2020.
- [24] Y. Bae, B. Cha, and J. Ryu, "Calibration and Evaluation for Visuo-haptic Collocation in Haptic Augmented Virtuality Systems," *International Journal of Control, Automation and Systems*, vol. 18, no. 5, pp. 1335–1342, 2020.
- [25] W. McNeely, "Robotic graphics: a new approach to force feedback for virtual reality," in *Proc. IEEE VRAIS*, 1993, pp. 336–341.
- [26] M. K. Rasmussen *et al.*, "Shape-changing interfaces: a review of the design space and open research questions," in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2012, pp. 735–744.
- [27] D. A. Bowman *et al.*, "Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments," *J. Visual Languages & Computing*, vol. 10, no. 1, pp. 37–53, 1999.
- [28] S. a. Haddadin, "Requirements for safe robots: Measurements, analysis and new insights," *Int. J. Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.
- [29] "Iso/ts 15066:2016 robots and robotic devices — collaborative robots," 2016.
- [30] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38–47, Aug. 2018.
- [31] V. R. Mercado, M. Marchal, and A. Lecuyer, "'Haptics On-Demand': A Survey on Encountered-Type Haptic Displays," *IEEE Transactions on Haptics*, pp. 1–16, 2021.