

OctoPocus in VR: Using a Dynamic Guide for 3D Mid-Air Gestures in Virtual Reality

Katherine Fennedy, Jeremy Hartmann, Quentin Roy, Simon Perrault, Daniel
Vogel

► **To cite this version:**

Katherine Fennedy, Jeremy Hartmann, Quentin Roy, Simon Perrault, Daniel Vogel. OctoPocus in VR: Using a Dynamic Guide for 3D Mid-Air Gestures in Virtual Reality. IEEE Transactions on Visualization and Computer Graphics, Institute of Electrical and Electronics Engineers, In press, 10.1109/TVCG.2021.3101854 . hal-03317991

HAL Id: hal-03317991

<https://hal.archives-ouvertes.fr/hal-03317991>

Submitted on 9 Aug 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.

OctoPocus in VR: Using a Dynamic Guide for 3D Mid-Air Gestures in Virtual Reality

Katherine Fennedy, Jeremy Hartmann, Quentin Roy, Simon Tangi Perrault, and Daniel Vogel

Abstract—Bau and Mackay’s OctoPocus dynamic guide helps novices learn, execute, and remember 2D surface gestures. We adapt OctoPocus to 3D mid-air gestures in Virtual Reality (VR) using an optimization-based recognizer, and by introducing an optional exploration mode to help visualize the spatial complexity of guides in a 3D gesture set. A replication of the original experiment protocol is used to compare OctoPocus in VR with a VR implementation of a crib-sheet. Results show that despite requiring 0.9s more reaction time than crib-sheet, OctoPocus enables participants to execute gestures 1.8s faster with 13.8% more accuracy during training, while remembering a comparable number of gestures. Subjective ratings support these results, 75% of participants found OctoPocus easier to learn and 83% found it more accurate. We contribute an implementation and empirical evidence demonstrating that an adaptation of the OctoPocus guide to VR is feasible and beneficial.

Index Terms—User Interfaces, Evaluation/methodology, Artificial, augmented, and virtual realities.

1 INTRODUCTION

GESTURES performed in the 3D space around the user without physical support, called “3D mid-air gestures”, really should be more common in Virtual Reality (VR). They match the nature of 6DOF controller input, they reduce the need for a space-occupying graphical interface, they increase the input vocabulary, and they facilitate expert performance: all desirable aspects for VR. Yet WIMP-inspired interactions, like pointing and “clicking” on graphical buttons and menus, are often used in VR, even though they are known to be more difficult without physical support or haptic feedback [2]. Perhaps the reason is that, unlike graphical widgets, gestures are not self-revealing by default [3], so users need to explicitly discover, learn, and memorize them. This goes against many usability guidelines [4] and even prompted Norman and Nielsen to characterize gestures as “a step backward” [5]. Solving this problem has been challenging for 2D gestures on desktops and touch surfaces, and grows even more difficult with the additional degrees-of-freedom (DOF) of mid-air gestures and challenges when visualizing 3D gesture paths.

A *guide* can help users discover and learn gestures. A basic form is a *crib-sheet* [6], where all available gestures are displayed as thumbnails with corresponding commands in a list or grid. This is a simple type of *feedforward*, since it displays what can be done and what the result will be [7, 8]. Feedforward in a crib-sheet is *static* since it does not change based on the context of a partially performed gesture, and it does not integrate *feedback* useful for learning other aspects of gesture performance, like system confidence

when recognizing gesture movements.

Marking Menus [9] and OctoPocus [1] are examples of *dynamic guides* for 2D gestures on desktops or tablets. Both filter feedforward outcomes and options based on a partially performed gesture, and both provide integrated recognition feedback. Marking menus are designed for gestures composed of straight lines with minimal feedback and simple feedforward, while OctoPocus works with gestures of arbitrary shapes with more continuous feedback and feedforward. Studies show that dynamic guides outperform static guides in speed and recall for 2D gestures [1, 10]. Since OctoPocus is more general, we examine how it can be adapted for 3D mid-air gestures in VR, and we compare it to a VR crib-sheet.

We note that Delamare et al. [11] implemented a 3D version of OctoPocus in a related but quite different context. They used a Kinect for 3DOF indirect input, a mouse button for delimiting the gesture, and a standard 2D desktop monitor to render 3D guide visualizations. Their main investigation focused on the effect of feedback and feedforward variations on the speed and movement accuracy of gesture performance. We adopt a key recommendation emerging from their work regarding the length of the feedforward gesture guide. However, their protocol did not measure gesture-to-command learning, recall, or guide usage rates, and they only compared OctoPocus to a video crib-sheet baseline in a 4-person preliminary study. The VR context is fundamentally different in terms of immersion, stereo rendering, and 6DOF controller input. It remains unclear if a video crib-sheet is better than the simpler thumbnail crib-sheet tested in the original Bau and Mackay’s OctoPocus study.

We adapt OctoPocus to 3D mid-air gestures in VR. We introduce a novel exploration mode that leverages VR 6DOF input to mitigate the challenges of 3D gesture guide visualization. Our system uses an iterative closest point (ICP) algorithm as a recognizer, which we modified to efficiently recognize partial 3D strokes suitable for the OctoPocus

- K. Fennedy and S.T. Perrault are with Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore. E-mail: katherine.fennedy@gmail.com and perrault.simon@gmail.com
- J. Hartmann, Q. Roy, and D. Vogel are with Cheriton School of Computer Science, University of Waterloo, Ontario, Canada. E-mail: jeremy.hartmann@uwaterloo.ca, quentin.roy@uwaterloo.ca, and dvoegel@uwaterloo.ca

Manuscript received Month DD, YYYY; revised Month DD, YYYY.

dynamic guide. The technique was evaluated in a near replication of Bau and Mackay’s study protocol using a VR crib-sheet as a baseline. This protocol measures more than execution speed and accuracy. Metrics like learning, recall, and guide usage rate are also captured.

Our results show participants were 14% more accurate with OctoPocus, 1.8s faster when executing the gesture, and found it easier to use. Our study replication also enables a direct comparison with Bau and Mackay’s OctoPocus in 2D. Our results are largely consistent, except Bau and Mackay found crib-sheet and OctoPocus to be comparable in terms of execution speed. In addition, while Bau and Mackay showed that OctoPocus improves gesture recall and incidental learning of non-practised gestures in 2D, we did not find evidence that these benefits translate to VR. Overall, our results demonstrate that OctoPocus is viable and even beneficial for VR mid-air gestures. However, its advantages over a simpler crib-sheet are not as pronounced as in a 2D interface.

Our main contributions are: (1) an enhanced design of OctoPocus for 3D mid-air gestures in VR; and (2) empirical results demonstrating the efficacy of OctoPocus in VR.

2 RELATED WORK

We describe how guidance can help users discover, perform, learn, and recall gestures, and we summarize previous gesture guidance work with a particular focus on methods that relate to 3D mid-air gesture guidance.

2.1 Gesture Guidance

In principle, a gesture does not require any visual interface: the gesture action is performed, and the associated command is triggered. Of course, this assumes the user already knows how to perform the gesture and its command mapping. A guide is a method to reveal possible gesture actions and their command mappings to users. This could be an offline user manual or tutorial video, but our focus is on visual guidance interaction techniques within the application interface.

The guidance technique itself can be initiated in many ways, such as pressing a graphical or physical button, or by pausing gesture execution for a moment. The latter is often referred to as “dwelling” or “press-and-hold.” Once triggered, the guidance information may share different spatial locality [12]: either displayed *adjacent* to the input area like a simple crib-sheet, or *collocated* near the input point to allow tracing like OctoPocus, or completely *separated* from the input area like OctoPocus3D [11]. A study revealed that systems that use a high level of guidance could maximize user performance while executing 2D gestures [10]. Each gesture may be visualized as a rendered line representing the ideal shape created by a gesture movement. This is sometimes referred to as a “template” since the line is based on an internal representation of the gesture path for the purpose of recognition. Other forms of visualizations include using arrows [13], user skeletons [14, 15, 16], hand shadows [17], body silhouettes [18], and “ghosted hints” [19].

Regardless of how and where the guide is displayed, it is a form of feedforward [7, 8] since it shows the possible

gesture options and outcomes. The feedforward can be static, meaning it is independent of any partially executed gesture input, or dynamic when the style and scope of the display continuously update based on user input for a partially completed gesture [10]. In addition, a guide may include feedback, which can also be static or dynamic. An example of static feedback is a display after gesture execution indicating what gesture was recognized, possibly with a level of confidence. On the other hand, dynamic feedback is provided during execution to communicate information like the current level of recognition confidence based on a partial gesture.

A related but tangential aspect of gesture learning is how to support a novice-to-expert transition [20]. This is especially important when multiple methods invoke a command, one designed for novices and another for experts. Suppose the two methods are quite different, such as graphical menus for novices and gestures for experts. In that case, there will be an initial drop in performance when users first adopt the expert method, even if that expert method is much faster when mastered in the long run [21]. We do not examine this aspect in our work.

2.2 Guidance for Mouse, Pen, and Surface Gestures

A crib-sheet is a common guide for mouse, pen, and surface gestures. It is a typically static display showing all gesture shapes and associated commands, typically in a list or grid. Past examples permanently display a crib-sheet in a toolbar [22], temporarily display a crib-sheet with a graphical button press [1, 6, 23, 24], or reveal it after dwelling for a second [17]. In addition, a crib-sheet can be enhanced to show each gesture’s text descriptions and animated renderings, such as in GestureBar [22]. Given the simplicity and predominance of a crib-sheet guide, it is a common baseline when evaluating more advanced guides.

Some guides combine characteristics of a crib-sheet with collocated visualizations and animations for a selected gesture. For example, ShadowGuides [17] is a technique for tabletop gestures composed of different hand postures and multi-finger movements. It uses a crib-sheet for initial gesture guide selection, then a dynamic feedforward via annotated shadow visualization showing how to perform the gesture in stages. Arpège [25] is a related guide for multi-finger chording, which dynamically reveals each finger to be laid down by the user.

A well-known example of an advanced gesture guide is the Marking Menu [9, 26], a circular menu of items selected using directional strokes. When an item is selected, a command is triggered, or another circular menu of sub-items is available using the same style of directional stroke movement. The design has a collocated feedforward triggered by a dwell, and the guide renders directional targets with labelled commands. It is dynamic because the guide for each menu level is shown incrementally after each directional gesture is drawn. Upon completion, feedback is provided as a beautified rendering of the ideal stroke. Several variations on the general approach have been proposed, including FlowMenu [27], Flower Menus [28], and Multi-Stroke Marking Menus [29, 30]. Studies have shown that radial layouts like Marking Menus are faster and more accurate than

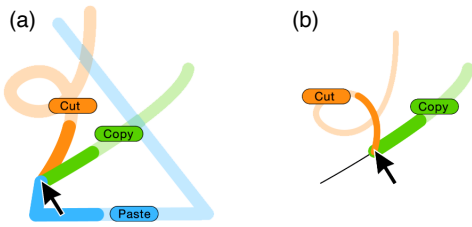


Fig. 1. Bau and Mackay's OctoPocus for 2D surface gestures: (a) when first triggered, all possible gesture paths are revealed; (b) as the 'Copy' gesture is partially traced out, the different 'Paste' gesture disappears and the similar, but less likely 'Cut' gesture path is rendered more transparent with a thinner line (figure from [1]).

conventional linear menus [31, 32], and radial layouts can exploit spatial memory by associating commands with cardinal orientations [33]. Marking Menus have been extended to touch [34] and mid-air [35, 36, 37] interactions. However, Marking Menus are limited to gestures composed of straight lines and do not support arbitrary gesture shapes such as curved paths, letters, or symbols.

Bau and Mackay's OctoPocus

A more general-purpose guidance technique is Bau and Mackay's OctoPocus [1]. Also activated with a dwell trigger, it renders collocated feedforward as possible gesture paths emanating from the current input location. Each path is coloured and labelled (Figure 1a), and they are dynamic: paths are filtered as the gesture is partially completed. Less likely ones fade away and eventually disappear while the most likely ones remain (Figure 1b). Continuous dynamic feedback is provided by how the relative thickness of the path lines communicates recognizer confidence.

Using a comprehensive experiment protocol, Bau and Mackay showed OctoPocus improves speed, accuracy, and recall compared to crib-sheet for arbitrary gestures and compared to Marking Menus for gestures composed of connected straight lines. OctoPocus was later integrated in a soft keyboard to guide gestural shortcuts [38] and was adapted to teach multi-finger chords [25]. Portions of the ShadowGuides whole-hand gesture guide are also based on OctoPocus [17]. Given the success of OctoPocus as a guide for arbitrary 2D gestures, we explore how it can be adapted to VR and use Bau and Mackay's protocol with a VR adapted crib-sheet baseline as a standard evaluation.

2.3 Guidance for Mid-air and Related Gestures

The additional dimension for movement makes 3D postures and gestures more complex to learn than their 2D counterparts. For mid-air hand gestures, real-time visual guidance and feedforward could be displayed directly on the hand using styled arrows [13] or on the screen of a smartphone held by the hand [39, 40]. In the context of motions in 3D space, it is common to have gestures that involve more than a hand. For example, a mirrored reflection of the user's skeleton [14, 15, 39] or a shadow of the user's body [18, 41] was used to guide whole-body movements in front of a large display. In both of these techniques, the feedforward guide is for a single predetermined movement, not to choose among possible movements. Therefore, their

effectiveness is limited to scenarios with a clear movement objective like dancing, yoga, and rehabilitation exercises. It makes sense why some guides were designed to focus on providing feedback on how far or close users' actions have been [42, 43] with only limited feedforward.

However, in our context of command selection tasks, the choice of gesture is on the user. We need feedforward strategies to communicate all available gestures, such that the corresponding guide can be revealed. For instance, Rovelo et al. [16] used a hierarchical tree structure to manually subdivide gestures into several body postures. While they demonstrated the feasibility of showing three simultaneous postures within a screen, it remains an open issue regarding how to scale feedforward to larger vocabularies of 3D mid-air gestures.

Delamare et al.'s "OctoPocus3D"

Delamare et al. [11] implemented OctoPocus for 3D mid-air hand gestures and used their system primarily to explore different feedback and feedforward mechanisms. In two main experiments, they measure how concurrent feedback, the portion of gestures displayed, and starting position of the guides affect completion time and recognition rate. Since their primary focus is on how participants use guidance, their experiment task only uses novice mode and their protocol does not test the learning or recall of gesture-to-command mappings. The most relevant result for our work is that rendering only the first 33% of the remaining guidance path made participants 8% faster than rendering all of the remaining path.

In a pilot study, Delamare et al. also showed that using their implementation of OctoPocus resulted in 30% more accurate gesture movements compared to video demonstrations. However, there were only 4 participants, execution speed was not reported, and learning or recall was not measured. In all studies, a Microsoft Kinect was used to track the hand, a mouse was used for a gesture trigger, and a desktop monitor to render all feedback and feedforward. Interaction in immersive VR with a high-precision 6DOF controller is fundamentally different from a 3D rendered scene on a desktop monitor with less precise 3DOF Kinect input (positional errors can be 1 to 3 cm [44]).

3 ADAPTING OCTOPOCUS FOR VR GESTURES

Our implementation of OctoPocus follows Bau and Mackay's original design [1] with the guidance design further justified by Delamare et al.'s results [11]. We further enhance OctoPocus for a VR environment with a new exploration mode, and our system enables efficient recognition using a refined recognition algorithm. In the description that follows, we adopt concise terminology used by previous work: "template" refers to the ideal gesture path rendered as feedforward, and "prefix" is an emphasized next portion along the ideal gesture path. Figure 2 provides a visual walk-through of the guide, and an accompanying video provides a full demonstration.

3.1 Guidance Visualization and Interaction

The trigger button of the VR controller is used to draw gestures, analogous to touching a surface or holding a

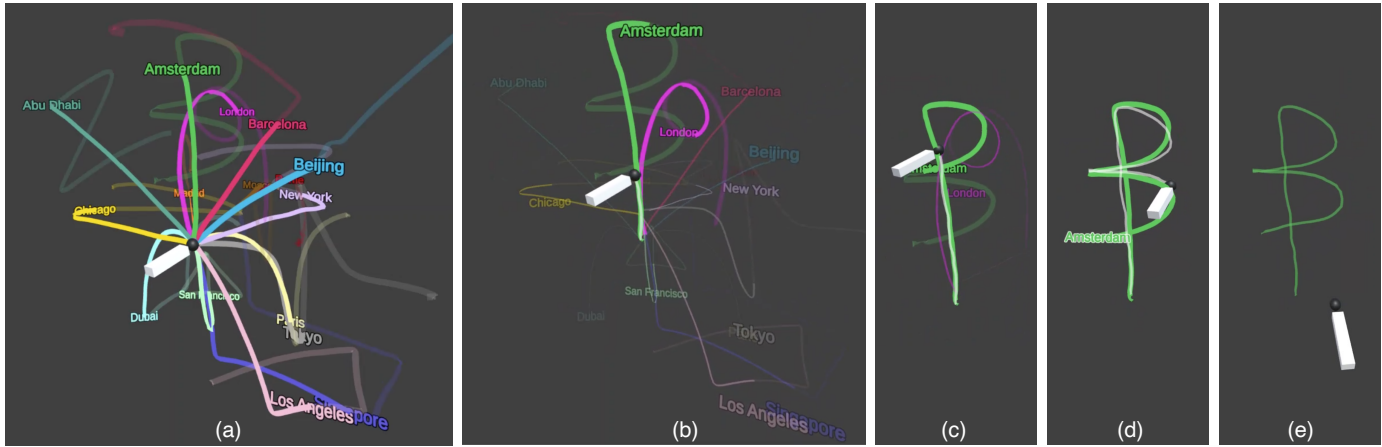


Fig. 2. VR OctoPocus guide walk-through: (a) when the trigger is first pressed, all 16 gestures are equally likely, so all feedforward ideal gesture paths are fully visible; (b) as the user draws, a white line shows the partially completed gesture path as feedback, and feedforward emphasizes the next portion of most likely gesture paths for ‘Amsterdam’ and ‘London’ “commands”, while less-likely paths become thinner and more transparent; (c) the feedforward path for ‘Amsterdam’ is now thicker and more opaque since it is more likely than ‘London’, all other paths have faded completely away since they are no longer possible; (d) the ‘Amsterdam’ label is highlighted in white to indicate the drawn gesture has met the minimum detectable threshold set by the recognizer; (e) when the trigger is released, all remaining guide paths disappear and terminal feedback shows the executed path in green indicating the ‘Amsterdam’ command was selected.

mouse button down in the original design. Like Bau and Mackay’s implementation and following rehearsal-based interfaces in general [45], the guide is activated after a short dwell time. Specifically, activation is when the trigger is held down and the controller position moves less than 7.5 mm during a window of 250ms. This is the same duration and comparable method to Bau and Mackay. Note that the guide may be triggered at any point, even if the gesture is partially executed.

A gesture can be executed without any guidance simply by moving at a reasonable speed without pausing. In that case, only a trace corresponding to the gesture would appear. We refer to this as “expert mode” since the gesture path and the command mapping must be memorized. The ultimate goal of any “novice” guide like OctoPocus is to help users learn the gesture set so they can use expert mode.

3.1.1 Path Visualization

When activated, the guide renders all possible gesture templates as coloured paths, each labelled with the associated command (Fig. 2a). All paths begin at the initial controller position. Each template begins with a fully opaque prefix, with a length equal to one-third of the template path length. The command label is rendered at the end of each prefix in the path colour with a thin black outline. Following Delamare et al.’s recommendation, the rest of the template is nearly transparent to reduce visual complexity [11].

In an actual application deployment, the orientation of the guide would appear relative to the user’s orientation. This could be relative to the user’s head orientation, controller orientation, or if available, their torso or shoulder orientation. However, choosing and implementing a specific body-centric orientation technique might introduce more variation across tasks and be less generalisable. For instance, it is not clear which is better, head orientation or controller direction. To simplify our implementation and to eliminate a potential confound in our experiment, we fixed the guide orientation to the virtual world and constrained participants

such that they faced the guide. This simulates a best-case body-centric orientation solution.

3.1.2 Feedback and Feedforward

Like the original version, our VR OctoPocus offers dynamic feedback and feedforward during partial execution of a gesture. Dynamic feedforward is accomplished by progressively filtering the template path visualization. As a specific gesture becomes less likely, the thickness and opacity of its template and label dynamically decrease (Fig. 2b), eventually disappearing when not possible at all (Fig. 2c). The opacity value of each template is determined by the calculation explained in Section 3.2. In addition, feedforward for the remaining possible paths is dynamically updated by sliding the emphasized prefix portion along the template, as if pushed by the controller. Feedback is in the form of path thickness to indicate recognition confidence, and the command label for the most likely path is highlighted in white when enough of the gesture is executed that recognition is possible (Fig. 2d). After the gesture is completed and the trigger is released, final feedback renders the actual gesture path in the recognized gesture colour for 500 ms (Fig. 2e).

3.1.3 Exploration Mode

The third spatial dimension in VR implies more overlap between templates, with gestures able to project backward and forward in the z-axis. Understanding the shape of these templates is more difficult than for surface gestures and requires greater exploration. We implemented an exploration mode triggered by pressing the controller grip button (different from the trigger button) after the OctoPocus guide has been revealed. This switches the orientation from world-relative to controller-relative, allowing the user to rotate and translate the visualization of all visible templates. Text labels, which represent what each gesture is mapped to, are always rendered facing the user. If the user was previously executing a gesture (by maintaining the trigger button), the drawing is paused while in exploration mode. While

this mode enables template exploration from any desired orientation, the gesture execution orientation is unchanged. As the user leaves the exploration mode by releasing the grip button, the guide snaps back to its original orientation with a 500ms animation and returns to draw mode at the new position of the controller.

3.2 Gesture Recognition

Recognizing partial gestures in 3D introduces additional challenges that are not present within a 2D space. Bau and Mackay’s 2D gesture recognizer was Rubine’s algorithm [46] with incremental turning angle representation [47]. They do not report specific recognition rates, but their overall accuracy measures of 93% and 95% suggest it worked well [1]. Delamare et al.’s 3D gesture recognizer used Dynamic Time Warping (DTW), but they report a low recognition rate, ranging between 47% and 81%. We chose to base our 3D gesture recognizer on a modified iterative closest point (ICP) algorithm [48, 49]. ICP can be considered a more general extension of the basic method used by the \$1 recognizer [50].

Extending and tuning ICP to be efficient for 3D mid-air gestures requires some important design choices. Even though only a single spatial dimension is added, the parameter space to find a match between the user’s path and the template is doubled for simple translation and rotation operations. In addition, the scale must also be considered. This results in 7-dimensional parameter space: 3 for rotation, 3 for translation, and 1 for uniform scale. For each instance the user is drawing, our process can be described in three phases.

Phase 1: Finding an optimal fit — For each template gesture, the drawn path and template are sub-sampled to reduce input complexity, similar to the initial stage of the \$1 recognizer [50]. These paths are then processed using an ICP algorithm on the constrained 7-dimensional parameter space. The constrained ranges are determined prior to optimization, and are tuned based on the needs of the template dataset. In our implementation, rotation is constrained to be within $\pm 22.5^\circ$, translation to be within $\pm 0.1m$, and scale to be within 0.2 to 2 times the original path size.

Phase 2: Calculating template weights — The root mean square error (RMSE) from phase one is used to calculate the weight (w) for for each template (k). The weights are calculated based on a simple radial basis function (RBF) Gaussian kernel [51]:

$$w_k = \exp\left(\frac{-RMSE^2}{\beta^2}\right). \quad (1)$$

This effectively converts the resulting RMSE into a weight between 0 and 1 for each template, with higher weights when errors are near zero. The parameter β is tuned to the template dataset and set at 15 by default.

Phase 3: Determining template likelihood — To find the likelihood (ℓ) of each template (k) based on the complete or partially complete user drawn path, the weights from the previous phase are normalized by their total sum, giving:

$$\ell_k = \frac{w_k}{\sum_i^N w_i}. \quad (2)$$

This likelihood determines whether that given template is the one the user is most likely drawing. An elegant property

of this is it allows us to determine equally likely templates for incomplete gestures and visualize them accordingly using the thickness or transparency in guidance paths.

4 EXPERIMENT

This experiment examines how our adaptation of OctoPocus for VR performs relative to a crib-sheet baseline using a near replication of Bau and Mackay’s protocol [1]. A blocked sequence of training and testing trials measures how well participants learn to execute 3D gestures and recall associations between gestures and city names, an abstract representation of command names. The accompanying video demonstrates all conditions and tasks.

4.1 Participants and Apparatus

We recruited 12 participants: ages 19 to 29 ($M = 24.8$, $SD = 3.4$); 5 identified as female and 7 identified as male; all were right-handed. Note that our experiment code supported left-handed participants. Handedness was not a recruiting criterion. In terms of VR experience, our participants can be considered novices: 2 use it twice, 6 use it once, and 4 have never used it. Each participant received \$15 for completing the 1.5-hour study.

As apparatus, we used an Oculus Quest VR head-mounted display (HMD) tethered to a Windows 10 laptop (2.6GHz Intel i7, GeForce RTX 2060). The techniques and experiment tasks were implemented as a Unity 2020.1.2f1 application. Participants were required to stand during experiment trials, and the virtual world environment was a uniform dark grey space.

4.2 Task and Stimuli

Before each trial, a sphere of diameter 20cm and labelled ‘Start’ is rendered at a fixed world location 30cm away from the user’s head and at the same height. This controls the participant’s position and body orientation. The participant moves the controller into the sphere and presses the ‘A’ button to begin the timed trial. Immediately, the sphere disappears and the target command name appears as white text with a 5cm line-height rendered 60cm away from the user.

The participant’s goal for the task is to perform a 3D gesture that matches the command name. There were two task variations. In a *training trial*, the participant could use a guide according to the experiment condition, either the OctoPocus technique (explained above) or a crib-sheet baseline (described below). If the participant chose to use the guide, the trial is marked as ‘novice’. Otherwise, it is marked as ‘expert’. In a *testing trial*, the participant could not use a guide. The timed portion of the trial ends when the participant releases the trigger after performing the gesture. In both training and testing trials, either the command corresponding to the performed gesture is displayed, or if the gesture motion did not meet a minimum threshold required by the recognizer, “not recognized” was shown. This visual feedback is positioned right below the target command name displayed since the beginning of the trial. Distinctive audio feedback is also provided as a “ding” or a soft buzzer sound.

There are three sets of 16 gestures each, one set for initial familiarization with the experiment task and techniques, and two sets used for measured trials in the main experiment. The first familiarization set used animal names, while the other two experimental sets used city names to represent associated gesture commands. These appeared in the guide and were used as the trial stimulus. The gestures were designed to be diverse in 3D shape by starting from Bau and Mackay’s basic 2D shapes and extending portions of the strokes along the z-axis directions. In addition, we maximized the spatial distribution of the gestures across x, y, and z axes and ensured no mnemonic mapping between the city name and gesture shape (e.g. no “P” for Paris).

4.3 Crib-Sheet Baseline

Following Bau and Mackay, we used a crib-sheet as a baseline guide comparison. This required some adaptation for VR usage, and our implementation works as follows. Prior to executing a gesture, the participant can press the controller grip button to view a 4 × 4 grid showing all gesture templates rendered in 3D (see Fig. 3). The command associated with each gesture is shown either under or above the template to minimize occlusion, and a small white sphere indicates the gesture start position. The templates are ordered alphabetically row-wise with start positions spaced 35cm horizontally and vertically within the grid. The centre of the grid is positioned at an offset of 10cm in the z-axis, away from the controller.

Similar to our OctoPocus exploration mode, the position and orientation of the grid are relative to the controller. It may be rotated and translated to observe the 3D templates from any desired angle. Like OctoPocus, each template and command label is displayed in an associated colour, and the participant’s gesture path changes from white to the associated colour when the trigger is released. Note that Bau and Mackay’s crib-sheet rendered all templates in black, but we believe this is a potential confound since colour can help visual search and retention [52, 53].

4.4 Procedure

We follow Bau and Mackay’s general procedure. After the participant provided information for demographics and related experience, the experimenter introduced the VR system and the experiment tasks and techniques. The main part of the experiment was completed in two sections, one per guide technique (OctoPocus and crib-sheet). We first start each section by asking participants to complete one training and one testing block with a gesture set of 16 animal names for about 10 minutes. Then, each section is followed by three phases: pre-test; main (a sequence of alternating training and testing sequences); and post-test. Figure 4 summarizes this design. Between each section, participants rested for up to 10 minutes to prevent accumulated fatigue. We used a different gesture set of 16 city names in each section to avoid confounding carryover effects. After both experiment sections were completed, the participant rated each guide technique on different aspects using a 7-point Likert scale: ease of learning, speed, accuracy, comfort, and ease of recall. Finally, the participant chose which technique they preferred overall.

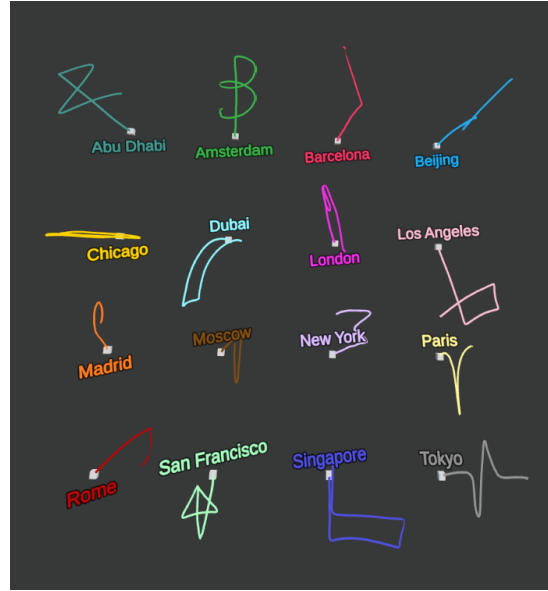


Fig. 3. Crib-sheet baseline condition: 16 gesture templates rendered in 3D in a 4x4 grid. The participant can rotate or translate the entire guide using the controller orientation and position.

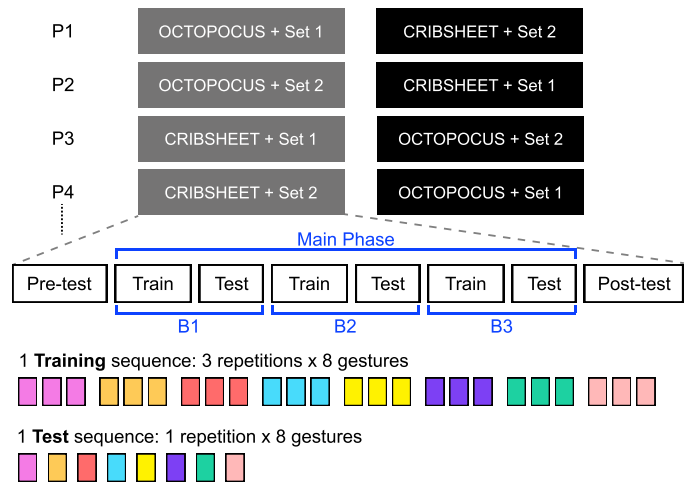


Fig. 4. Experimental Design. Participants started each technique with a Pre-test phase of testing trials, followed by three blocks of separated training and testing trials, then a final sequence of post-test testing trials. In Pre- and Post-test phases, all 16 gestures in the gesture set were tested. In the main phase blocks, only 8 out of 16 gestures were used as stimulus and all participants received the same 8 gestures.

4.4.1 Pre-test Phase

The purpose of this phase is to verify the gesture sets used for each technique are unknown without “guessable” command mappings, and to establish a relative recall baseline. The participant completed 16 testing trials, one for each gesture in the set (i.e. without the guide or correctness feedback) in random order. This is the first time they see these gestures and mappings, so they are instructed to provide their best guess. Like Bau and Mackay, we expect a low (close to 0%) accuracy.

4.4.2 Main Phase

The purpose of this phase is to measure guide usage, gesture performance, and gesture mapping learning. Note that the

participant is trained and tested on only 8 gestures out of the 16 gestures in the set, but the guide displays all 16 gestures. There are three blocks, each with a sequence of training trials followed by a sequence of testing trials. At the start of each block, the participant is reminded to learn as many gestures as possible because they will be tested at the end of the block.

Each *training sequence* has a total of 24 training trials, 8 gestures each with 3 repetitions. Gestures are ordered randomly but repeated three times consecutively for rehearsal. The training portion of each block measures how the guide is used and how gestures are performed in terms of speed and accuracy.

Each *test sequence* has 8 testing trials, one for each of the 8 gestures they are trained on. Since test trials disable the guide, the participant has to rely entirely on their memory. Note that Bau and Mackay did not include a final testing sequence at the end of the third block. We include this to more consistently measure recall for the 8 training gestures. The test portion of each block evaluates recall performance.

4.4.3 Post-test Phase

The purpose of this phase is to re-test recall for the 8 practised gestures and incidental learning of the 8 other gestures in the set that were not practised. The participant completes 16 testing trials, one per gesture presented in random order. Participants were not told that the other 8 gestures would be tested until the beginning of the post-test phase.

4.5 Design

We used a within-subject design with GUIDE as a primary independent variable with 2 levels { OCTOPOCUS, CRIBSHEET }, and gesture SET and main phase BLOCK as secondary independent variables. The order of presentation of GUIDE was fully counterbalanced (see Figure 4). SET has 2 levels { SET1, SET2 }, where the same set was paired with a different GUIDE over alternating participants to avoid any confounding effect of SET on GUIDE. BLOCK has 3 levels { B1, B2, B3 }. In the main phase, each participant completes 144 training trials: 2 GUIDES \times 3 BLOCKS \times 8 gestures \times 3 repetitions. Each participant completes 112 testing trials across all phases: 2 GUIDES \times [16 pre-test gestures + (3 main phase BLOCKS \times 8 gestures) + 16 post-test gestures].

Like Bau and Mackay, our dependent variables include trial time, overall execution accuracy, recall, incidental learning, and subjective ratings. We use modified naming conventions for some measures to improve readability, and we introduce additional dependent variables to expand our analysis. These differences are explained below.

5 RESULTS

We identified 40 trials (2.3%) as outliers since they had *Trial Time* more than 3 standard deviations from the mean of all training trials. We also verified there were no main effects or interactions for SET and guide ORDER on any time or accuracy measures. For analysis, trials were first aggregated by participant and factors used in the test. When the assumption of sphericity was violated, we corrected both p-values and degrees of freedom using Greenhouse-Geisser

($\epsilon < 0.75$). If data residuals were found to be non-normal by Shapiro-Wilk, we name the specific data transformation used to correct normality for analysis. A GUIDE \times BLOCK ANOVA with pairwise t-tests with Bonferroni corrections for post hoc comparisons were used for objective continuous measures. Wilcoxon Signed Rank tests were used to analyze subjective ratings. Similar to [54], we also report 95% confidence intervals as [low, high] for the difference of means to minimize the over-interpretation of p-values [55].

5.1 Time

We examine three related time measures calculated from training trials marked as ‘novice’ (*i.e.* when the participant uses either guide). *Trial Time* is from when the target command label appears until the participant releases the trigger to complete the gesture. Bau and Mackay called this measure ‘overall input time’, and unlike Bau and Mackay, we only include correct novice trials during training. *Reaction Time* is the beginning period of a trial before the participant reacts with explicit input: measured from when the target command label appears until the participant first presses a controller button to view the OctoPocus or crib-sheet guide. This is the same as Bau and Mackay’s ‘access time’. *Input Time* is the remaining period of the trial, from when the participant started viewing the guide until they release the trigger to complete the gesture. Note that *Trial Time* is the sum of *Reaction Time* and *Input Time*.

Overall, OctoPocus has a longer *Reaction Time* but less *Input Time* compared to the crib-sheet. Both *Trial Time* and *Input Time* decreased after the first training block for both techniques.

5.1.1 Trial Time

OctoPocus and crib-sheet are similar in terms of overall trial time (Figure 5a). There was no main effect of GUIDE on log transformed *Trial Time* ($F_{1,11} = 2.27, p = .161, \eta_G^2 = .03$), and there was no interaction between GUIDE and BLOCK. Mean times across all blocks: OCTOPOCUS ($M = 8.1s$) and CRIBSHEET ($M = 9.0s$) ($[-0.3s, 2.2s]$). However, we found a significant main effect of BLOCK ($F_{2,22} = 10.14, p < .001, \eta_G^2 = .11$) on *Trial Time*. Pairwise comparisons show that participants were slower in B1 ($M = 9.7s$) as compared to B2 ($M = 8.1s, p < .01, \Delta = -1.6s, [-3.1s, -0.2s]$) and B3 ($M = 7.7s, p < .01, \Delta = -2.0s, [-3.5s, -0.5s]$).

5.1.2 Reaction Time

OctoPocus was on average 0.9s slower than the crib-sheet to view the guide before starting to draw the gesture (Figure 5b). There was a significant main effect of GUIDE ($F_{1,11} = 68.68, p < .00001, \eta_G^2 = .52$) on *Reaction Time*, with OCTOPOCUS ($M = 2.0s$) requiring more time to react than CRIBSHEET ($M = 1.1s, \Delta = -0.9s, [-1.1s, -0.7s]$). There was no main effect of BLOCK on *Reaction Time* ($F_{2,22} = 0.39, p = .682, \eta_G^2 = .01$) and no interaction between GUIDE and BLOCK.

5.1.3 Input Time

OctoPocus required 1.8s less time to draw gestures than crib-sheet, with times decreasing after the first block for both guides (Figure 5c). There was a significant main effect of

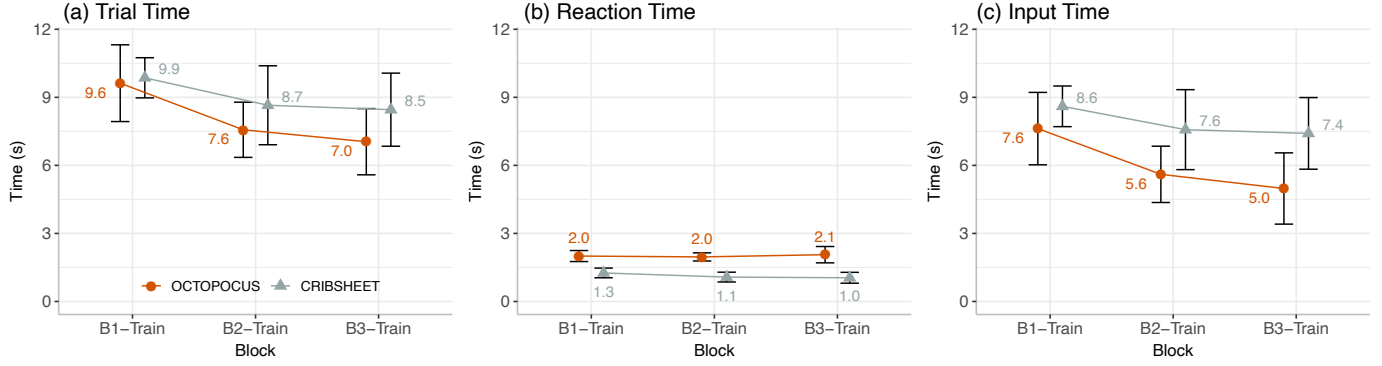


Fig. 5. Time measurements by training BLOCK for each GUIDE for: (a) *Trial Time*, (b) *Input Time*, and (c) *Reaction Time* (error bars in all graphs represent 95% confidence intervals). Note that *Trial time* is the sum of *Reaction Time* and *Input Time*.

GUIDE on square-root transformed *Input Time* ($F_{1,11} = 7.81$, $p = .017$, $\eta_G^2 = .14$) showing OCTOPOCUS ($M = 6.1s$) is less than CRIBSHEET ($M = 7.9s$, $\Delta = 1.8s$, $[0.5s, 3.1s]$). There was also a main effect of BLOCK ($F_{2,22} = 13.05$, $p < .001$, $\eta_G^2 = .12$) on *Input Time*. Pairwise comparisons show that participants were slower in B1 ($M = 8.1s$) as compared to B2 ($M = 6.6s$, $p < .01$, $\Delta = -1.5s$, $[-3.0s, 0s]$) and B3 ($M = 6.2s$, $p = .032$, $\Delta = -1.9s$, $[-3.5s, -0.4s]$). We did not find any interaction between GUIDE and BLOCK.

5.2 Execution Accuracy

We examine three related measures of execution accuracy also using only training trials. We use “accuracy” to create a positive measure instead of an equivalent negative measure of “error rate”. We use the term “execution” because the participant should know the target gesture since they can use the guide or they are confident performing the gesture without the guide. This means this measure represents how well they can execute the gesture shape to match the template. Therefore, *Overall Execution Accuracy* is the extent to which gestures drawn by a participant match the gesture template targets during a single block. Bau and Mackay called this measure ‘overall error rate’, and they did not also decompose it into two groups based on the trial mode as we do. We also define *Expert Execution Accuracy* for trials without guidance, and *Novice Execution Accuracy* for trials with guidance.

Overall, OctoPocus is more accurate than crib-sheet for all three measures. In addition, *Expert Execution Accuracy* decreased in the last two blocks.

5.2.1 Overall Execution Accuracy

OctoPocus was more accurate than crib-sheet by 13.8% (Figure 6a). There was a significant main effect of GUIDE ($F_{1,11} = 71.99$, $p < .00001$, $\eta_G^2 = .41$) on *Reaction Time*, with OCTOPOCUS ($M = 90.2\%$) being more accurate than CRIBSHEET ($M = 76.4\%$, $\Delta = -13.8\%$, $[-17.7\%, -9.8\%]$). There was no main effect of BLOCK on *Overall Execution Accuracy* ($F_{2,22} = 0.39$, $p > .05$) and no interaction between GUIDE and BLOCK.

5.2.2 Expert Execution Accuracy

When participants do not use the guide at all, OctoPocus was more accurate than crib-sheet by 10.5%, with accuracy

of both guides worsening between the last two blocks (Figure 6b). There was a significant main effect of GUIDE ($F_{1,11} = 24.87$, $p < .001$, $\eta_G^2 = .18$) on *Expert Execution Accuracy*, with OCTOPOCUS ($M = 87.3\%$) more accurate than CRIBSHEET ($M = 77.0\%$, $\Delta = -10.3\%$, $[-16.1\%, -4.6\%]$). There was also a significant main effect of BLOCK ($F_{2,22} = 4.28$, $p = .027$, $\eta_G^2 = .13$) on *Expert Execution Accuracy*. Pairwise comparisons show that participants were more accurate in B2 ($M = 86.9\%$) as compared to B3 ($M = 76.4\%$, $p < .01$, $\Delta = -10.5\%$, $[-17.6\%, -3.5\%]$). There was no interaction between GUIDE and BLOCK.

5.2.3 Novice Execution Accuracy

When participants use the guide, OctoPocus was more accurate than crib-sheet by 20.2% (Figure 6c). There was a significant main effect of GUIDE ($F_{1,11} = 24.90$, $p < .001$, $\eta_G^2 = .27$) on *Novice Execution Accuracy*, with OCTOPOCUS ($M = 93.4\%$) being more accurate than CRIBSHEET ($M = 73.2\%$, $\Delta = -20.2\%$, $[-28.2\%, -12.0\%]$). There was no main effect of BLOCK on *Novice Execution Accuracy* ($F_{2,22} = 0.13$, $p > .05$) and no interaction between GUIDE and BLOCK.

5.3 Learning

We measure how well participants learned gestures by examining the execution accuracy at different testing trials. Note that no guide is available in testing trials, so participants must either guess or rely on what they learned during any training trials they completed. *Recall* measures the accuracy of executing the 8 gestures practised during training, and it is measured during the main phase of the experiment. “*Recall*” during the pre-test phase measures performance without any prior knowledge, which is helpful to verify that the gesture sets were not easily guessable. *Incidental Learning* measures the accuracy of the remaining 8 gestures not practised during training, and it is measured during the post-test phase. Bau and Mackay assessed learning at similar phases but used ‘error rates’ instead.

5.3.1 Recall

During the main phase, OctoPocus and crib-sheet are comparable in terms of *Recall* (Figure 7a), and participants learned gestures at a similar rate. There was no significant main effect of GUIDE ($F_{1,11} = 0.02$, $p > .05$) on *Recall*.

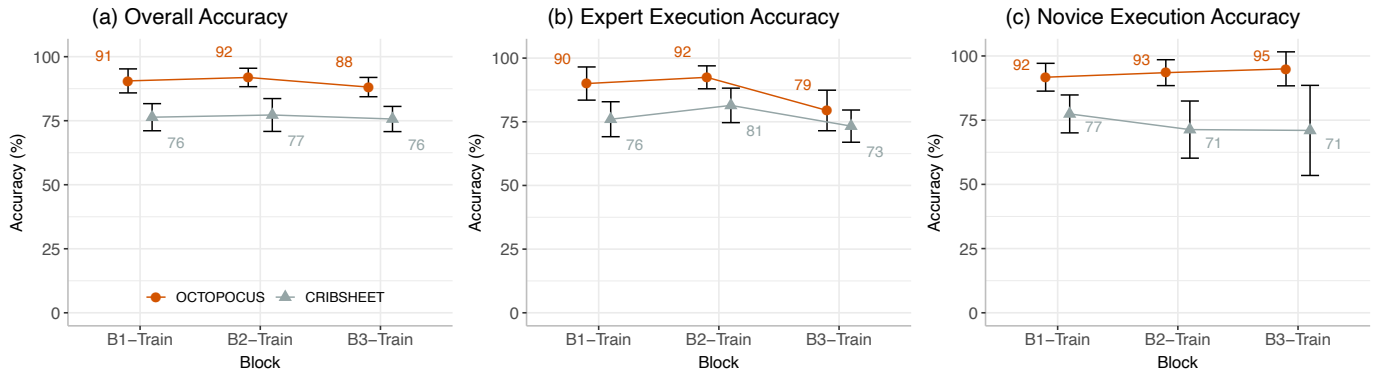


Fig. 6. Execution Accuracy by training BLOCK for each GUIDE for: (a) all trials, (b) expert trials only, and (c) novice trials only.

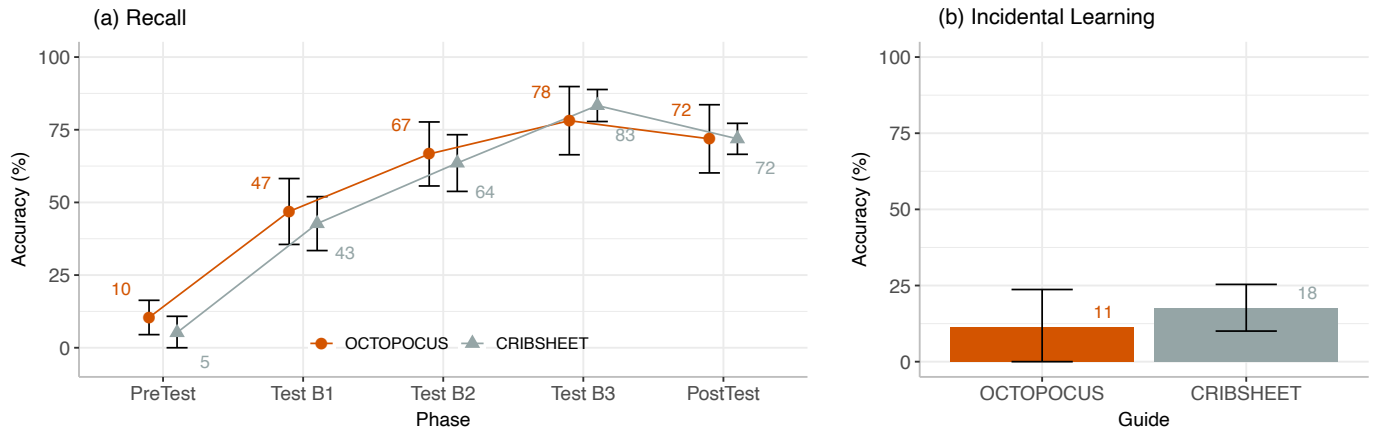


Fig. 7. (a) *Recall* on 8 practised gestures across all three phases and across BLOCK during the training phase; and (b) *Incidental Learning* on 8 non-practised gestures by GUIDE during post-test phase.

Mean accuracy of OCTOPOCUS is 63.9% while CRIBSHEET is 63.2% ([-11.4%, 10.0%]). There was a significant main effect of BLOCK ($F_{2,22} = 31.13, p < .00001, \eta_G^2 = .43$) on *Recall*. Pairwise comparisons show that participants could recall more gestures after each training block: recall at B1 ($M = 44.8%$) increased at B2 ($M = 65.1%, p < .001, \Delta = 20.3%, [9.8%, 30.8%]$), and increased again at B3 ($M = 80.7%, p < .001, \Delta = 15.6%, [5.7%, 25.6%]$). We did not find any interaction between GUIDE and BLOCK.

We can consider the pre-test and post-test phases as additional blocks of testing trials and examine recall across these five “blocks”. With both guides, participants could recall more gestures after the first block of training compared to the pre-test, and their recall did not significantly change between the last main phase test block and the post-test phase. There was no main effect of GUIDE ($F_{1,11} = 0.17, p > .05$) on *Recall*, but there was a significant main effect of BLOCK ($F_{4,44} = 94.81, p < .0001, \eta_G^2 = .74$). Pairwise comparisons show that recall differences are mostly significant ($p < .0001$) except that *Recall* was similar ($p > .05$) between main phase’s B3 ($M = 80.7%$) and post-test phase ($M = 65.1%, [-18.1%, 0.4%]$). We did not find any interaction between GUIDE and BLOCK.

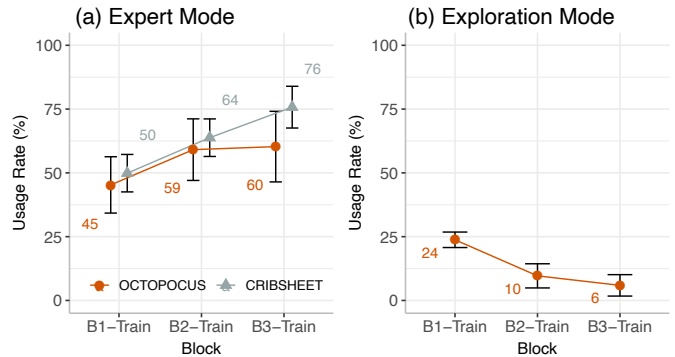


Fig. 8. Usage rates for (a) *Expert Mode* and (b) *Exploration Mode* during training BLOCKS for each GUIDE.

5.3.2 Incidental Learning

During the post-test phase, OctoPocus and crib-sheet are also comparable in terms of the number of gestures they learned that they were not explicitly trained on (Figure 7b). There was no significant main effect of GUIDE ($F_{1,11} = 0.73, p > .05$), participants learned an average of 0.9 other gestures with OCTOPOCUS ($M = 11.5%$) and 1.4 items with CRIBSHEET ($M = 17.7%, [-21.7%, 9.2%]$).

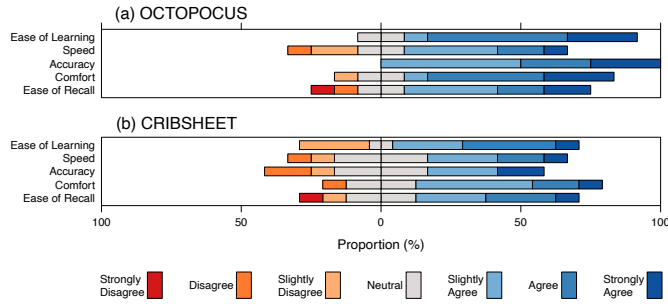


Fig. 9. Responses to each subjective measure for each guide.

5.4 Expert Mode Usage

Expert Mode Usage Rate is the ratio of trials performed without any guide to the total number of training trials during a main phase block. Bau and Mackay measured the inverse, novice mode usage rate; we focus on expert mode since this is the goal of a rehearsal-based interface like OctoPocus. Overall, OctoPocus and crib-sheet are similar in terms of *Expert Mode Usage Rate* (Figure 8a). There was no significant main effect of GUIDE ($F_{1,11} = 2.25, p > .05$) with participants using expert mode with OCTOPOCUS ($M = 54.9\%$) at a comparable rate with CRIBSHEET ($M = 63.2\%$, $[-1.1\%, 17.6\%$]). As in several other measures, we found a significant main effect of BLOCK ($F_{2,22} = 12.70, p < .001, \eta_G^2 = 0.19$). Pairwise comparisons show that participants used expert mode more often in B1 ($M = 47.6\%$) as compared to B2 ($M = 61.5\%$, $p < .001, \Delta = 13.9\%$, $[4.0\%, 23.7\%]$) and B3 ($M = 68.1\%$, $p < .001, \Delta = 20.5\%$, $[9.4\%, 31.5\%]$). There was no interaction between GUIDE and BLOCK.

5.5 OctoPocus Exploration Mode Usage

Participants used exploration mode 23.8% of the time in the first block, but dropped to slightly less than 6% in the third block. There was a significant main effect of BLOCK ($F_{2,22} = 21.41, p < .001, \eta_G^2 = 0.48$) on *Exploration Mode Usage Rate*. Pairwise comparisons show that usage rate in B1 ($M = 23.8\%$) is less than B2 ($M = 9.6\%$, $p < .01, \Delta = -14.2\%$, $[-20.1\%, -8.1\%]$) and B3 ($M = 5.9\%$, $p < .01, \Delta = -17.9\%$, $[-23.4\%, -12.3\%]$).

5.6 Subjective Ratings

Each participant rated different aspects of each guide using a 7-point Likert scale, which forms these measures: *Ease of Learning*, *Speed*, *Accuracy*, *Comfort*, and *Ease of Recall*. Exact question phrasing is provided in the supplementary material. *Ease of Learning* focuses on how effective the guide was for learning commands and associated gesture movements during training and *Ease of Recall* captured how well the guide prepared the participant for testing trials. *Speed*, *Accuracy*, and *Comfort* capture different aspects of executing gestures with the help of the guide.

The ratings are comparable for all measures except ease of learning and accuracy for which OctoPocus was rated more favourably than crib-sheet (Figure 9). There was a significant effect of GUIDE ($Z = 10.0, p = .03$) on *Ease of Learning*, where 9 out of 12 participants rated OCTOPOCUS

more favourably than CRIBSHEET. The median rating for OCTOPOCUS is 6 while the median rating for CRIBSHEET is 5. There was also a significant effect of GUIDE ($Z = 9.5, p = .02$) on *Accuracy*, where 10 out of 12 participants rated OCTOPOCUS more favourably than CRIBSHEET. The median rating for OCTOPOCUS is 5.5 while the median rating for CRIBSHEET is 4.

Each participant was also asked to choose OctoPocus or crib-sheet as their overall favourite, and 7 out of 12 chose OctoPocus.

6 DISCUSSION

We first discuss overall patterns in our experiment results, and then compare them against Bau and Mackay’s original OctoPocus as well as Delamare et al. [11]’s work.

6.1 Our Evaluation of OctoPocus for VR

Our results revealed several advantages when using OctoPocus as a guide in VR, but also that a simpler crib-sheet can be as good or better in some cases.

Time trade-off between OctoPocus and crib-sheet

OctoPocus and crib-sheet guides resulted in similar overall trial times, but analysis of reaction and input portions of a trial revealed differences. Participants activated the crib-sheet 0.9s faster, but participants performed gesture input 1.8s faster after using the OctoPocus guide. We believe the increased speed of gesture input with OctoPocus is because the guide enables participants to directly map their input movement to the collocated template visualization. The slower reaction time for OctoPocus can partially be attributed to how it is only revealed after pausing once drawing starts. This not only adds 250ms for the dwell time, but it also requires more time to initially prepare to draw as the controller is moved to a central comfortable position. With crib-sheet, there is no delay and the guide is activated with a button independent of the drawing state. Some participants asked if they could “view the guide before drawing” [P3,7] in OctoPocus because “the act of pressing the trigger button feels like committing to an action” [P7]. This disadvantage for OctoPocus may disappear over time since participants quickly switched from novice to expert mode (Figure 8a).

OctoPocus leads to more accurate gestures

Participants were more accurate with OctoPocus than with crib-sheet during gesture execution. The trend is the same whether the guide is used in novice trials or after gestures are learned in expert trials: OctoPocus is 10.3 to 20.2 percentage points more accurate (Figure 6). This difference was noticed by participants, who rated OctoPocus more highly for accuracy (Figure 9). Participants commented that OctoPocus enabled them to “build a spatial map” [P11] through its feedforward mechanism, further shown by 7 participants who commented OctoPocus helped them to “trace paths accurately”. On the other hand, participants found crib-sheet less accurate because they had to “draw from memory” [P2,4], which likely resulted in the spatial mismatch between target and drawn gestures.

OctoPocus and crib-sheet comparable for learning

While we did not observe a difference between OctoPocus and crib-sheet for recall, we noted that recall increased across blocks of test trials to reach an average of 80.7% during the last block (Figure 7a), suggesting that participants learned gestures quickly. This effect is corroborated by the proportion of expert trials during training trials, which increased from an average of 47.6% in the first block to 68.1% in the last block (Figure 8a). By the end of training, more than 2 out of 3 trials are done in expert mode on average.

Both OctoPocus and crib-sheet support incidental learning

Participants were able to learn gestures that they did not practice: 14.6% of gestures or 1.2 new gestures out of 8 (Figure 7b). Participants commented that “*OctoPocus teaches (them) more gestures implicitly because when a gesture is being drawn, the neighbouring gestures will appear near it*” [P6] and that crib-sheet allows them to “*quickly glance across many gestures*” [P1,3,7]. The incidental learning greatly varied across participants. One participant [P4] learned 6 out of the 8 unpractised gestures with OctoPocus, while two participants [P6,P9] learned 3 new gestures and 4 participants learned 2 new gestures with either technique.

Visualization benefits and challenges

The general pattern of our results shows OctoPocus was as good or better in almost all measures, suggesting there are benefits to a collocated dynamic visualization for a 3D gesture guide. Our special OctoPocus “*exploration mode*” was also used by participants, especially during initial training. We believe this helped to mitigate some of the challenges of visualizing multiple collocated 3D gesture paths. However, there are still challenges. Some participants commented that OctoPocus creates “*one big lump of lines*” [P12] that “*looks complicated*” [P3] and “*congested, as everything is everywhere*” [P10], making it “*harder to find the labels*” [P3,P4,P6]. This led some to speculate that OctoPocus may be less desirable [P4,P6] if more commands need to be displayed.

Although the subjective ratings did not show a clear preference for crib-sheet on any measure, some participants commented that its organized grid presentation enabled them to “*quickly glance across many gestures*” [P1,P3,P7]. This may suggest a hybrid approach combining OctoPocus with a crib-sheet where different groups of gestures are clustered in a grid, and the collocated gesture group is dynamic, perhaps shifting smoothly to the controller position as it becomes more likely. This could be by group, or even a cross-group filter that moves most likely gesture templates to the collocated controller position. Another approach could be to make “*explore mode*” more like a crib-sheet, with the gesture templates spread out for this initial visualization. A related approach is to use another 3D mid-air gesture input feature to dynamically select among groups of collocated gestures. For example, using different hand postures to filter out groups of templates, reminiscent of Freeman et al.’s tabletop guidance system [17].

6.2 Comparison to Bau and Mackay’s 2D OctoPocus

Our near replication of Bau and Mackay’s experiment design enables us to contrast and compare the performance of

OctoPocus and crib-sheet when gestures are 3D in VR and 2D on a desktop.

OctoPocus vs. crib-sheet

Like our study, Bau and Mackay found OctoPocus significantly more accurate than a crib-sheet baseline, suggesting that OctoPocus is better at training people to perform more precise gesture movements in 2D and 3D. However, Bau and Mackay also found overall trial time for OctoPocus lower than crib-sheet, while we observed no overall difference in trial time, there is a trade-off between reaction time and input time. Bau and Mackay found no difference between conditions for input time or reaction time. In addition, Bau and Mackay found a significantly higher post-test recall for OctoPocus, but this appears to be because recall for our crib-sheet condition (83%) was much better than Bau and Mackay’s (57%).

These different results across studies may be because our crib-sheet interaction design was more optimized for activation speed and recall cues. Just like OctoPocus, our crib-sheet guide could be activated anywhere using the controller grip, but Bau and Mackay’s crib-sheet required moving the mouse across the interface and clicking on a button. Just like OctoPocus, our crib-sheet guide rendered gesture templates using distinct and memorable colours, but Bau and Mackay’s crib-sheet rendered all templates in black and white. We made these guide-independent design choices to make crib-sheet as equivalent to OctoPocus as possible. We suspect that the crib-sheet tested in the Bau and Mackay study introduced interaction design confounds that led to significant differences in time and recall.

3D gestures are harder to execute

Overall, OctoPocus in 2D seems faster than 3D (5.7s vs. 8.1s for trial time) and more accurate (96% vs. 90.2% for accuracy in training). Notably, OctoPocus was used with comparable frequency in 2D (40%) and 3D (45%), so these differences are not due to guide usage. These differences are likely due to the additional complexity of performing and remembering a 3D gesture. For example, the VR input context uses larger, less precise drawing motions with a 6DOF controller in mid-air compared to smaller, precise, and physically supported drawing motions with a 2DOF mouse. Based on McMahan [56]’s Framework for Interaction Fidelity Analysis (FIFA), user performance does not necessarily increase with higher interaction fidelity [57]. For example, a study conducted by Bhargava et al. [58] demonstrated that in some motor skill training simulations, mid-fidelity conditions performed worse than the relatively lower and higher fidelity conditions. This evidence supports our results, and it suggests that future interaction researchers explore lower fidelity alternatives to support the execution of 3D gestures in VR.

3D gestures are harder to learn

Like Bau and Mackay, we observed a significant trend over blocks of reduced time for training trials and increased recall for testing trials. However, the final post-test recall and incidental learning rates appear quite different. Bau and Mackay only report a combined post-test “*error rate*”

of 27% (i.e. 73% “recall”) that includes both the final recall of the eight trained items and incidental learning rates of the other eight untrained items. We can estimate the incidental learning rate by considering that 73% of 16 total items means 11.7 items were recalled. Further assuming that all 8 trained items were perfectly recalled, we can estimate at least 3.7 items were recalled by incidental learning. This is substantially higher than incidental learning of 0.9 items in our study. We believe this is due to 3D gestures being much harder to learn overall and the increased visual complexity of the 3D OctoPocus visualization.

6.3 Comparison to Delamare et al.’s “OctoPocus3D”

Delamare et al.’s 3D OctoPocus implementation and study designs are very different, and they provide few details about their video crib-sheet baseline condition. However, with these limitations in mind, we can still make some meaningful comparisons. First, Delamare et al.’s OctoPocus required between 10s and 17s (an estimated 11s on average) to complete a trial, which is 36% longer than the mean time for our OctoPocus technique (8.1s). Second, Delamare et al. report recognition rates of 55%-81% for OctoPocus across all studies and 47% for video crib-sheet in their preliminary pilot with only 4 participants. This appears lower than our overall accuracy measures of 90% for OctoPocus and 76% for our static crib-sheet. This suggests that our ICP recognizer is more reliable than Delamare et al.’s DTW method. These differences support the contribution of our work in designing and testing a VR version of OctoPocus, but we note that Delamare et al.’s focus on feedback and feedforward methods was different.

6.4 Limitations

Unrecognized gestures

For any gesture recognition system, there is a subtle interplay between what the user thinks the gesture is, how the system interprets them, and what its actual representation is as a ground truth. This is considerably more apparent in 3D than in 2D, as the user has an extra degree of freedom when externalizing an internal representation of a gesture they just learned. Our recognizer used a database of hand-drawn templates which may account for some of the false-negatives that occurred since there may have been a disconnect between the hand-drawn template and the participants’ interpretation of it. Supplementing each template with additional samples could further improve the detection of these edge cases and improve the overall accuracy of the gesture recognition system.

Arbitrary gesture shapes and mappings

The execution and recall of a gesture may have been affected by the diverse 3D gesture path shapes and the arbitrary mapping to city names. Initially, some participants found it “challenging to build a mental mapping” [P7,11,12] between the gestures and names of each city because we intentionally avoided explicit mnemonic associations, just like Bau and Mackay’s study. However, with practice, participants were “able to recall better with gestures that have a weird or unique shape” [P6,10]. This is supported by a study [59] whose

evidence suggests that leveraging “desirable difficulty” can improve learning. In general, there is a need to balance the familiarity and complexity of 3D gestures, both individually and as a holistic set.

Recognition feedback during testing

As explained earlier, test trials provided visual and audio feedback about the recognition and correctness of the drawn gesture. One might question if the presence of such feedback provided a hint for subsequent trials within the same block. We investigated this potential bias by examining all 1,344 testing trials for cases where an incorrect gesture was drawn, and the recognized gesture was a match for a future test trial in the same block. This identified 52 (3.9%) correct trials where participants might have benefited from feedback on prior incorrect trials, but also 39 (2.9%) incorrect trials where participants fail to leverage the possible benefit from prior incorrect trials. The small and similar counts in both cases suggest that feedback in testing trials did not have any pronounced effect on recall accuracy.

OctoPocus label occlusion

We designed the gesture set in such a way that the templates emanate as evenly as possible into different directions, which helped minimize label occlusion globally. This could be further improved by automatically testing for occlusions with other labels or gesture templates and making any needed local refinements in label positions. Some participants noted OctoPocus labels could be “harder to find [than in crib-sheet] because some may overlap with other labels or gestures” [P5,10]. Occlusion could be further optimized using view management techniques like the Hedgehog labelling approach proposed by Tatzgern et al. [60]. In addition, we plan to investigate how OctoPocus can be adapted to hierarchical groupings of gestures into menus in 3D, and how labels of multiple commands belonging to the same category could be merged into one. For example, labels for “pear” and “strawberry” could be merged into “fruit”.

7 CONCLUSION

In this paper, we leveraged a design recommendation from Delamare et al. [11] in our adaptation of Bau and Mackay’s OctoPocus [1] original design for an immersive VR environment. To better understand the performance, we compared it to a 3D adaptation of a crib-sheet for VR and replicated Bau and Mackay’s experiment protocol. Our results suggest that when executing gesture movements, participants were faster and more accurate with OctoPocus. Both guidance techniques were comparable for learning gestures and incidental learning, with participants remembering gestures they were not trained on. Comparing our implementation with the original 2D OctoPocus, we find that 3D interaction is more complex, both in terms of execution and visualization, which explains why participants were faster and more accurate in 2D OctoPocus.

While our specific implementation and study are for VR, 3D mid-air gestures can also be used as input in other settings, like large displays and spatial augmented reality (SAR) [61]. In these contexts, the lack of a personal stereoscopic 3D for display may be challenging, but we

believe that OctoPocus could also work in these contexts with appropriate modifications and optimizations. For instance, a SAR setting could rely on multiple 2D planes like surrounding walls and floors to simulate 3D, or on-body projections like LightGuide [13] to render dynamic feedforward for multiple gestures simultaneously.

While our results show that OctoPocus is an effective guide for 3D mid-air gestures in VR, the good performance of crib-sheet suggests that simpler and lower fidelity guidance should still be considered. We hope this will lead to more VR interfaces exploiting the power of gestures and fully leverage the input and visualization affordance of VR.

ACKNOWLEDGMENTS

This work was supported by NSERC Discovery Grant 2018-05187, the Canada Foundation for Innovation Infrastructure Fund 33151 “Facility for Fully Interactive Physio-digital Spaces”, Ontario Early Researcher Award ER16-12-184, and the Singapore Ministry of Education and Singapore University of Technology and Design (SUTD) Start-up Research Grant (T1SRIS18141).

REFERENCES

- [1] O. Bau and W. E. Mackay, “Octopocus: A dynamic guide for learning gesture-based command sets,” in *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 37–46. [Online]. Available: <https://doi.org/10.1145/1449715.1449724>
- [2] A. Van Dam, “Post-wimp user interfaces,” *Communications of the ACM*, vol. 40, no. 2, pp. 63–67, 1997.
- [3] T. Baudel and M. Beaudouin-Lafon, “Charade: Remote control of objects using free-hand gestures,” *Commun. ACM*, vol. 36, no. 7, p. 28–35, Jul. 1993. [Online]. Available: <https://doi.org/10.1145/159544.159562>
- [4] J. Nielsen and R. Molich, “Heuristic evaluation of user interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’90. New York, NY, USA: ACM, 1990, pp. 249–256. [Online]. Available: <http://doi.acm.org/10.1145/97243.97281>
- [5] D. A. Norman and J. Nielsen, “Gestural interfaces: A step backward in usability,” *Interactions*, vol. 17, no. 5, p. 46–49, Sep. 2010. [Online]. Available: <https://doi.org/10.1145/1836216.1836228>
- [6] G. Kurtenbach, T. P. Moran, and W. Buxton, “Contextual animation of gestural commands,” *Computer Graphics Forum*, vol. 13, no. 5, pp. 305–314, 1994. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.1350305>
- [7] T. Djajadiningrat, K. Overbeeke, and S. Wensveen, “But how, donald, tell us how? on the creation of meaning in interaction design through feedforward and inherent feedback,” in *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, ser. DIS ’02. New York, NY, USA: Association for Computing Machinery, 2002, p. 285–291. [Online]. Available: <https://doi.org/10.1145/778712.778752>
- [8] J. Vermeulen, K. Luyten, E. van den Hoven, and K. Coninx, “Crossing the bridge over norman’s gulf of execution: Revealing feedforward’s true identity,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1931–1940. [Online]. Available: <https://doi.org/10.1145/2470654.2466255>
- [9] G. Kurtenbach and B. Buxton, “User learning and performance with marking menus,” in *CHI ’94*, ser. CHI ’94. New York, NY, USA: ACM, 1994, pp. 258–264. [Online]. Available: <http://doi.acm.org/10.1145/191666.191759>
- [10] F. Anderson and W. F. Bischof, “Learning and performance with gesture guides,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1109–1118. [Online]. Available: <https://doi.org/10.1145/2470654.2466143>
- [11] W. Delamare, T. Janssoone, C. Coutrix, and L. Nigay, “Designing 3d gesture guidance: Visual feedback and feedforward design options,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 152–159. [Online]. Available: <https://doi.org/10.1145/2909132.2909260>
- [12] W. Delamare, C. Coutrix, and L. Nigay, “Designing guiding systems for gesture-based interaction,” in *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 44–53. [Online]. Available: <https://doi.org/10.1145/2774225.2774847>
- [13] R. Sodhi, H. Benko, and A. Wilson, “Lightguide: Projected visualizations for hand movement guidance,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 179–188. [Online]. Available: <https://doi.org.proxy.lib.uwaterloo.ca/10.1145/2207676.2207702>
- [14] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice, “Youmove: Enhancing movement training with an augmented reality mirror,” in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 311–320. [Online]. Available: <https://doi.org/10.1145/2501988.2502045>
- [15] C. Ackad, A. Clayphan, M. Tomitsch, and J. Kay, “An in-the-wild study of learning mid-air gestures to browse hierarchical information at a large interactive public display,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1227–1238. [Online]. Available: <https://doi.org/10.1145/2750858.2807532>
- [16] G. Rovelo, D. Degraen, D. Vanacken, K. Luyten, and K. Coninx, “Gestu-wan - an intelligible mid-air gesture guidance system for walk-up-and-use displays,” in

- Human-Computer Interaction – INTERACT 2015*, J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds. Cham: Springer International Publishing, 2015, pp. 368–386.
- [17] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor, “Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures,” in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 165–172. [Online]. Available: <https://doi.org/10.1145/1731903.1731935>
- [18] R. Walter, G. Bailly, and J. Müller, “Strikeapose: Revealing mid-air gestures on public displays,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 841–850. [Online]. Available: <https://doi.org/10.1145/2470654.2470774>
- [19] S. White, L. Lister, and S. Feiner, “Visual hints for tangible gestures in augmented reality,” in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 47–50.
- [20] A. Cockburn, C. Gutwin, J. Scarr, and S. Malacria, “Supporting novice to expert transitions in user interfaces,” *ACM Comput. Surv.*, vol. 47, no. 2, Nov. 2014. [Online]. Available: <https://doi.org/10.1145/2659796>
- [21] J. Scarr, A. Cockburn, C. Gutwin, and P. Quinn, “Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 2741–2750. [Online]. Available: <https://doi.org/10.1145/1978942.1979348>
- [22] A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola, “Gesturebar: Improving the approachability of gesture-based interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 2269–2278. [Online]. Available: <https://doi.org/10.1145/1518701.1519050>
- [23] K. Hinckley, S. Zhao, R. Sarin, P. Baudisch, E. Cutrell, M. Shilman, and D. Tan, “Inkseine: In situ search for active note taking,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 251–260. [Online]. Available: <https://doi.org/10.1145/1240624.1240666>
- [24] S. J. Castellucci and I. S. MacKenzie, “Graffiti vs. unistrokes: An empirical comparison,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 305–308. [Online]. Available: <https://doi.org/10.1145/1357054.1357106>
- [25] E. Ghomi, S. Huot, O. Bau, M. Beaudouin-Lafon, and W. E. Mackay, “Arpège: Learning multitouch chord gestures vocabularies,” in *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 209–218. [Online]. Available: <https://doi.org/10.1145/2512349.2512795>
- [26] G. Kurtenbach and B. Buxton, “The limits of expert performance using hierarchic marking menus,” in *INTERACT '93 and CHI '93*, ser. CHI '93. New York, NY, USA: ACM, 1993, pp. 482–487. [Online]. Available: <http://doi.acm.org/10.1145/169059.169426>
- [27] F. Guimbretiére and T. Winograd, “Flowmenu: Combining command, text, and data entry,” in *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 213–216. [Online]. Available: <https://doi.org/10.1145/354401.354778>
- [28] G. Bailly, E. Lecolinet, and L. Nigay, “Flower menus: A new type of marking menu with large menu breadth, within groups and efficient expert mode memorization,” in *Proceedings of the Working Conference on Advanced Visual Interfaces*, ser. AVI '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 15–22. [Online]. Available: <https://doi.org/10.1145/1385569.1385575>
- [29] S. Zhao and R. Balakrishnan, “Simple vs. compound mark hierarchical marking menus,” in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 33–42. [Online]. Available: <https://doi.org/10.1145/1029632.1029639>
- [30] S. Zhao, M. Agrawala, and K. Hinckley, “Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1077–1086. [Online]. Available: <https://doi.org/10.1145/1124772.1124933>
- [31] P. M. Fitts, “The information capacity of the human motor system in controlling the amplitude of movement.” *Journal of experimental psychology*, vol. 47, no. 6, p. 381, 1954.
- [32] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman, “An empirical comparison of pie vs. linear menus,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 95–100. [Online]. Available: <https://doi.org/10.1145/57167.57182>
- [33] G. Bailly, E. Lecolinet, and L. Nigay, “Visual menu techniques,” *ACM Comput. Surv.*, vol. 49, no. 4, Dec. 2016. [Online]. Available: <https://doi.org/10.1145/3002171>
- [34] J. Zheng, X. Bi, K. Li, Y. Li, and S. Zhai, *M3 Gesture Menu: Design and Experimental Analyses of Marking Menus for Touchscreen Mobile Interaction*. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–14. [Online]. Available: <https://doi.org/10.1145/3173574.3173823>
- [35] G. Ren and E. O'Neill, “3d marking menu selection with freehand gestures,” in *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, 2012, pp. 61–68.
- [36] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen, “Extended pie menus for immersive virtual

- environments," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 4, pp. 644–651, 2013.
- [37] W. Li, X. Wan, Y. Shi, N. Yao, C. Wang, and Z. Gao, "Depth and breadth of pie menus for mid-air gesture interaction," *International Journal of Human-Computer Interaction*, vol. 37, no. 2, pp. 131–140, 2021.
- [38] J. Alvina, C. F. Griggio, X. Bi, and W. E. Mackay, "Commandboard: Creating a general-purpose command gesture input space for soft keyboard," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 17–28. [Online]. Available: <https://doi.org/10.1145/3126594.3126639>
- [39] A. Kamal, Y. Li, and E. Lank, "Teaching motion gestures via recognizer feedback," in *Proceedings of the 19th International Conference on Intelligent User Interfaces*, ser. IUI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 73–82. [Online]. Available: <https://doi.org/10.1145/2557500.2557521>
- [40] J. Henderson, S. Mizobuchi, W. Li, and E. Lank, "Exploring cross-modal training via touch to learn a mid-air marking menu gesture set," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: <https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3338286.3340119>
- [41] F. Alt, S. Geiger, and W. Höhl, "Shapelineguide: Teaching mid-air gestures for large interactive displays," in *Proceedings of the 7th ACM International Symposium on Pervasive Displays*, ser. PerDis '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi.org/10.1145/3205873.3205887>
- [42] M. Sousa, J. a. Vieira, D. Medeiros, A. Arsenio, and J. Jorge, "Sleevear: Augmented reality for rehabilitation using realtime feedback," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, ser. IUI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 175–185. [Online]. Available: <https://doi.org/10.1145/2856767.2856773>
- [43] M. Dürr, R. Weber, U. Pfeil, and H. Reiterer, "Eguide: Investigating different visual appearances and guidance techniques for egocentric guidance visualizations," in *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 311–322. [Online]. Available: <https://doi.org/10.1145/3374920.3374945>
- [44] A. Mobini, S. Behzadipour, and M. Saadat Foumani, "Accuracy of Kinect's skeleton tracking for upper body rehabilitation applications," *Disability and Rehabilitation: Assistive Technology*, vol. 9, no. 4, pp. 344–352, jul 2014.
- [45] B. Lewis and D. Vogel, "Longer delays in rehearsal-based interfaces increase expert use," *ACM Trans. Comput.-Hum. Interact.*, vol. 27, no. 6, Nov. 2020. [Online]. Available: <https://doi.org/10.1145/3418196>
- [46] D. Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, p. 329–337, Jul. 1991. [Online]. Available: <https://doi.org/10.1145/127719.122753>
- [47] G. Iannizzotto and L. Vita, "A multiscale turning angle representation of object shapes for image retrieval," in *Visual Information and Information Systems*, D. P. Huijsmans and A. W. M. Smeulders, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 609–616.
- [48] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, feb 1992. [Online]. Available: <http://ieeexplore.ieee.org/document/121791/>
- [49] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, apr 1992. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/026288569290066C>
- [50] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes," in *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*. New York, New York, USA: ACM Press, 2007, p. 159. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1294211.1294238>
- [51] M. J. D. Powell, "Algorithms for approximation," J. C. Mason and M. G. Cox, Eds. New York, NY, USA: Clarendon Press, 1987, ch. Radial Basis Functions for Multivariable Interpolation: A Review, pp. 143–167. [Online]. Available: <http://dl.acm.org/citation.cfm?id=48424.48433>
- [52] C. Jansson-Boyd, N. Marlow, and M. Bristow, "The influence of colour on visual search times in cluttered environments," *Journal of Marketing Communications*, vol. 10, pp. 183–193, 09 2004.
- [53] X. Zhuang and T. V. Pappathomas, "Cue relevance effects in conjunctive visual search: Cueing for location, color, and orientation," *Journal of Vision*, vol. 11, no. 7, pp. 6–6, 2011.
- [54] I. K. Duncan, S. Tingsheng, S. T. Perrault, and M. T. Gastner, "Task-based effectiveness of interactive contiguous area cartograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2136–2152, 2021.
- [55] G. Cumming, *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge, 2013.
- [56] R. P. McMahan, "Exploring the effects of higher-fidelity display and interaction for virtual reality games," Ph.D. dissertation, Virginia Tech, 2011.
- [57] R. P. McMahan, D. A. Bowman, D. J. Zielinski, and R. B. Brady, "Evaluating display fidelity and interaction fidelity in a virtual reality game," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 4, pp. 626–633, 2012.
- [58] A. Bhargava, J. W. Bertrand, A. K. Gramopadhye, K. C. Madathil, and S. V. Babu, "Evaluating multiple levels of an interaction fidelity continuum on performance and learning in near-field training simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24,

no. 4, pp. 1418–1427, 2018.

- [59] C. Diemand-Yauman, D. M. Oppenheimer, and E. B. Vaughan, “Fortune favors the (): Effects of disfluency on educational outcomes,” *Cognition*, vol. 118, no. 1, pp. 111–115, 2011. [Online]. Available: <https://doi.org/10.1016/j.cognition.2010.09.012>
- [60] M. Tatzgern, D. Kalkofen, R. Grasset, and D. Schmalstieg, “Hedgehog labeling: View management techniques for external labels in 3d space,” in *2014 IEEE Virtual Reality (VR)*, 2014, pp. 27–32.
- [61] O. Bimber and R. Raskar, *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005.



Daniel Vogel is an Associate Professor in the School of Computer Science at the University of Waterloo. His research area is Human Computer Interaction, focusing on fundamental characteristics of human input and novel forms of interaction for current and future computing form factors.



Katherine Fennedy received the B.Eng. degree in Information Systems Technology and Design from the Singapore University of Technology and Design (SUTD), in 2016. She is currently working towards a PhD degree in SUTD, with research interests including interaction techniques in the context of modes, command selection tasks and virtual reality environment.



Jeremy Hartmann is a PhD candidate at the School of Computer Science at the University of Waterloo. His research interests focus on virtual and augmented reality within Human Computer Interaction.



Quentin Roy is a postdoctoral fellow at the University of Waterloo. He studies how people interact with their computing systems, and how computing systems can be leveraged to augment human capabilities. His research focuses on complex activities that require the use of elaborate software integrating diverse functions. He has contributed to reduce human errors, extend user capabilities, and facilitate critical users tasks.



Simon Tangi Perrault obtained his PhD degree in computer science from Telecom ParisTech (France) in 2013. He is an Assistant Professor in Information Systems Technology and Design at the Singapore University of Technology and Design (SUTD). Previously, he was a Visiting Professor at the Korean Advanced Institute of Science and Technology (Korea), and Assistant Professor at Yale-NUS College (Singapore).