Designing 3D Gesture Guidance: Visual Feedback and Feedforward Design Options

William Delamare, Thomas Janssoone, Céline Coutrix and Laurence Nigay

Université Grenoble Alpes, LIG, CNRS
F-38000 Grenoble, France

{William.Delamare, Celine.Coutrix, Laurence.Nigay}@imag.fr, janssoot@gmail.com

ABSTRACT

Dynamic symbolic in-air hand gestures are an increasingly popular means of interaction with smart environments. However, novices need to know what commands are available and which gesture to execute in order to trigger these commands. We propose to adapt OctoPocus, a 2D gesture guiding system, to the case of 3D. The OctoPocus3D guidance system displays a set of 3D gestures as 3D pipes and allows users to understand how the system processes gesture input. Several feedback and feedforward visual alternatives are proposed in the literature. However, their impact on guidance remains to be evaluated. We report the results of two user experiments that aim at designing OctoPocus3D by exploring these alternatives. The results show that a concurrent feedback, which visually simplifies the 3D scene during the execution of the gesture, increases the recognition rate, but only during the first two repetitions. After the first two repetitions, users achieve the same recognition rate with a terminal feedback (after the execution of the gesture), a concurrent feedback, both or neither. With respect to feedforward, the overall stability of the 3D scene explored through the origin of the pipes during the execution of the gestures does not influence the recognition rate or the execution time. Finally, the results also show that displaying upcoming portions of the gestures allows 8% faster completion times than displaying the complete remaining portions. This indicates that preventing visual clutter of the 3D scene prevails over gesture anticipation.

CCS Concepts

• **Human-centered computing~Gestural input** • Human-centered computing~Displays and imagers.

Keywords

3D hand gesture; Guidance; Feedback; Feedforward.

1. INTRODUCTION

Interaction through 3D in-air hand gestures has gained much attention due to novel hardware capabilities [8,26]. However, users still have to know (1) what commands are available and (2) how to trigger them. This is a significant bottleneck in the acceptance of 3D gesture interaction by a wide public [17]. We address this problem by extending a 2D gesture guiding system, namely OctoPocus [4], to 3D gestures. OctoPocus3D displays gestures as on-screen 3D colored pipes to allow users to discover and execute available 3D gestures.

Taxonomies of 3D gestures [1,12] allow us to clarify the gestures we aim to guide with our OctoPocus3D guide. We focus on semaphoric or symbolic gestures that define a vocabulary to trigger discrete commands. Semaphoric gestures can be static (postures) or dynamic (motion paths in 3D). For instance, drawing a triangle in mid-air triggers the 'play' command. We address such *semaphoric* and *dynamic* gestures, since they are commonly used for interacting with smart environments [13] and users need to guess and/or learn the gestures [1,12]. We hence address the difficulty for novices to learn and perform 3D semaphoric dynamic gestures as it is a substantial bottleneck that is limiting the progress of dynamic and symbolic in-air gestures in smart environments.

Facing the difficulty of users to guess, learn and perform semaphoric dynamic gestures, several studies explored the design of gesture sets for novices. A recurring finding from these studies is that there is no 3D gesture set (user-defined [13] or expert-defined [16]) with a perfect consensus among users [13,25]. One solution includes reusing users' own gestures when interacting with a new system [26]. However, this solution requires systems to be compatible with each other, which is not likely to occur soon. As a consequence, it is still difficult for novices to discover (1) which commands are available, (2) what is the gesture corresponding to a particular command and (3), how to perform the gesture.

While recent progress has been made for guiding 2D surface gesture sets [4,9] or a single 3D gesture [2,23], guiding a set of 3D gestures is still an open issue. Indeed, to the best of our knowledge, there is no guiding system designed for the guidance of a complete set of a wide range of dynamic semaphoric 3D hand gestures. Unlike 2D gesture set [4] or single 3D gesture guidance [23], the display of several 3D paths cannot be co-localized with the user's hand, unless users are intrusively instrumented for stereoscopic display of gesture paths. We therefore study design guidance based on an additional display, distant from the location of the performed 3D gestures. Since 2D gestures are commonly represented with 2D lines as in OctoPocus [4], we display 3D gestures as 3D pipes. Having defined the visual presentation of the gestures, our research question then lies in exploring the dynamicity of the visual presentation. We thus study the design options that are related to feedback and feedforward mechanisms in the literature.

- *Feedback* provides information about the past actions, e.g., a trace displaying the performed gesture and/or recognition scores. Such information aims at letting users correct their gestures. On the one hand, several guidance systems provide such feedback at the termination phase after the gesture is finished [5,11,18]. On the other hand, feedback can be concurrent, i.e. during the continuation phase, while the gesture is being performed [2,4,9]. In particular, the concurrent

AVI '16, June 07-10, 2016, Bari, Italy
© 2016 ACM. ISBN 978-1-4503-4131-8/16/6…$15.00.
DOI: http://dx.doi.org/10.1145/2909132.2909260

feedback introduced by OctoPocus [4] and reused in other studies [3,9] intends to let the user understands how the system and its gesture recognizer process the input gesture at run time.

- *Feedforward* provides information relevant to the actions the user is going to perform in order to trigger a command. Feedforward therefore includes the available commands and the corresponding gesture paths. For OctoPocus3D, we study two complementary aspects. First, the guide can represent the complete remaining portions of the gestures to be performed [3,4] or only upcoming portions of them [15,23]. These options influence the visual complexity of the resulting 3D scene. Second, the guide can display the paths in a relative way (i.e. position coupled to the user's hand) [2,4] or in an absolute way (i.e. position anchored in space independently of the current position of the user's hand) [3,18]. These options influence the stability of the 3D scene.

We first review related work and describe design alternatives based on existing feedback and feedforward mechanisms. We then report and discuss results from two user experiments exploring the impact of these design alternatives on the completion time and recognition rate with OctoPocus3D. Results show that concurrent feedback leads to a 10% higher recognition rate than that obtained with no feedback only during the first two repetitions. Thus, OctoPocus3D can include a concurrent feedback for novices and then removes it in order to reduce its negative influence on motor learning. In addition, displaying portions of gestures leads to 8% faster completion time than displaying complete remaining portions of the gestures. This result shows that for rapidity, the overall simplicity of the scene prevails over the gesture anticipation enabled by displaying the complete gestures. Finally, displaying absolute or relative paths does not impact performances during guidance, showing that the overall instability of the 3D scene does not impact on the execution of gestures with OctoPocus3D.

2. RELATED WORK

We do not aim at exhaustively describing the 46 existing guiding systems [6]. Rather, we review the design options for the feedback and feedforward mechanisms. We illustrate these options with a subset of existing 2D and 3D guiding systems [6].

2.1 Feedback: Terminal and Concurrent

Feedback can be terminal or concurrent. GestureBar [5] provides a terminal feedback after training a gesture (i.e. "nice job" or "not quite right"). Another example [11] consists of an arrow showing the recognition score on a 3-scale graphical bar. Other guiding systems display the recognized command [19].

Another solution to provide terminal feedback consists of a visual comparison of the executed gesture and the intended one. It is important to note that such terminal feedback mechanisms can only occur when the system knows which gesture the user intended to execute, for instance during a training session. For example in [11] the system displays a representation of a teacher and of the user. Based on this representation and after the execution of a gesture, the terminal feedback is a replay mechanism of both performed and intended gestures. Another system to perform Tai-Chi movements [18] uses a similar terminal feedback, enriched with statistical information.

The system guiding Tai-Chi movement [18] also provides concurrent feedback. The guide continuously maps the measured error to the brightness of a line displayed between the user's hands on screen. With similar avatars, the solution in [22] uses arrows as concurrent feedback to notify a deviation between the user's avatar and the teacher's avatar (feedforward). Other systems provide concurrent feedback to guide several gestures. These guiding systems are often called "dynamic" [4]. For instance, OctoPocus [4] and ShadowGuides [9] display gestures with colored 2D lines labeled with command names. Once the user executes a gesture, the current recognition scores of the gestures (feedback) are mapped on the thickness of the lines, i.e. the less likely to be recognized, the thinner the line. Such concurrent feedback aims at letting users know how the system processes the input so that users can correct their gesture during the execution. In addition, once a score is below a threshold, the system does not display the corresponding gesture anymore. Such a concurrent feedback simplifies the on-screen visual complexity during the execution of a gesture. This option also suits large gesture sets, in order to simultaneously reveal all available gestures while guiding the user.

Other systems like YouMove [2] that guides 3D, full body movements, do not provide any feedback during the 'Movement Guide' stage. This is also the case of other 3D gesture guiding systems, focusing on data collection related to gesture set design [13], gesture recognizers [8,10], and physical therapy [23].

Previous studies also showed that concurrent feedback may have a negative effect on the accuracy of motor learning if users rely too much on it during practice [3,21]. However, solutions exist for reducing this negative effect. For instance, the Adaptive Guide [3] does not suffer from this negative effect since the concurrent feedback disappears, together with the feedforward, according to the experience of the user. This prevents the user from relying too much on the guide. Thus, since (1) we focus on novice users and (2), adaptive solutions exist to reduce the impact of concurrent feedback, we consider concurrent feedback.

2.2 Feedforward: Portion of Gestures

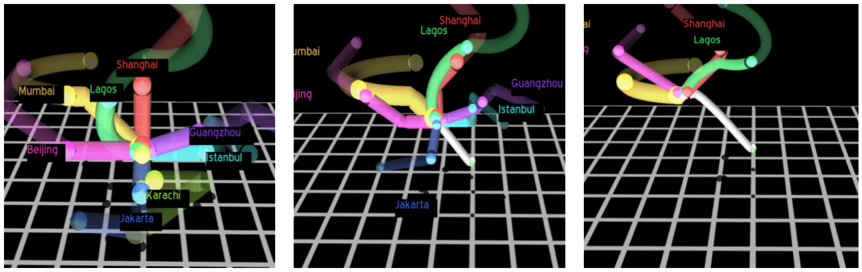
The feedforward can represent the complete remaining portions of the gestures to be performed. Examples include OctoPocus [4], the crib sheet of [3] or Marking Menu [14]. Other examples include gestures that are presented completely, but one at a time, using on-screen videos [8,10] or animations [23]. These options, in particular for 3D hand gestures, can decrease the legibility of the 3D scene, but can also increase the user's ability to anticipate the execution of the gesture.

Others only represent the upcoming portions of the gestures. This is the case for the Interactive Crib Sheet [15], the Hierarchical Marking Menu [15], the Gestu-Wan [20] and Lightguide [23]. Gestu-Wan [20] proposes to guide upper body and hand gestures through the decomposition of gestures into a sequence of key postures. However, such decomposition might be difficult to adapt to our case with any 3D hand gestures. Indeed, gestures need to be manually decomposed, which might be difficult and cumbersome for curved gestures. Lightguide [23] proposes different visual cues to render the upcoming direction of a single gesture, projected onto the user's hand (e.g., a spot or a 3D arrow). These options can increase the legibility of the 3D scene, but can also decrease the user's ability to anticipate the execution of the gesture.

In previous work, even though both options have been considered, the impact of the quantity of information regarding the portion of the gestures that is displayed has not been evaluated.

2.3 Feedforward: Origin of the Guide

In the literature, the guide can display the gestures in a relative way. In this case, the position of the guide is coupled to the user's hand. Examples include the dynamic guide of [3], the Marking Menu [14], OctoPocus [4] and YouMove [2]. In the later case, the



(a) (b) (c)

Figure 1: Illustration of the concurrent feedback and feedforward. The feedforward mechanism displays the remaining portion of gesture paths from the digital representation of the hand. (a) Original scene. (b) User starts to follow the gesture toward the direction up-left. Radius of unlikely recognized gestures (on the right of the scene) have decreased (concurrent feedback). (c) Gestures with predicted score below a threshold have vanished (concurrent feedback). During the entire execution of the gesture, 3D pipes start from the digital hand's position.

guides are coupled to each body joints. Lightguide [23] provides collocated 3D guidance, projected onto the user's hand. In this case, the number of gestures that can be revealed is very limited.

Other guides display the gestures in an absolute way. In this case the position is anchored in space independently of the current position of the user's hand. This is the case of the static guide of [3]. A static crib sheet is another example, which usually lists the name of the commands and illustrates gestures with 2D drawings. Other examples include gestures that are presented one by one to users using on-screen videos [8,10] or animations [23]. This is also the case of the guide of Tai-Chi movements [18] using a teacher avatar seen from an absolute third-person point-of-view.

In previous work, even though both options have been considered, their impact on the interaction has not been evaluated. Anderson and Bischof [3] did not separate the effect of the static origin of the guides and the concurrent feedback in their study. They evaluated the learning aspect of 2D gestures with four guiding systems:

1. A static crib-sheet guide displayed in the upper-left corner of the screen;
2. A dynamic guide;
3. A static guide: similarly to the dynamic guide, the static guide displays gestures with lines, except that (1) the feedforward is not impacted by the feedback mechanism (i.e. at any time, the guide displays the complete remaining paths of all gestures) and (2), the starting position of gesture paths remains at the initial location of the pen;
4. An adaptive guide: the Adaptive Guide provides the same feedforward as the static guide, but disappears sooner and sooner as the user becomes an expert.

If we focus on their training results (i.e. users as novices), the static guide condition allows a better accuracy than all the other guiding systems. But it is not clear which of the two parameters impact the accuracy of the guiding system: the absence of concurrent feedback and/or the static origin of the guides. These two parameters, concurrent feedback and starting position of the guides, remain to be evaluated, for 2D gestures as well as for 3D gestures. It is difficult to state if the stability of the 3D scene provided by an absolute origin of the guide is more, less or equally important than the unique location of attention provided by a guide displayed relatively to the user's hand.

To sum up, existing systems consider different feedback and feedforward mechanisms. Their impact on the interaction remains to be evaluated. The feedback and the feedforward options provide several alternatives for presenting the gestures to the user. We propose to evaluate these alternatives using OctoPocus3D, a guide we designed to present 3D gestures with on-screen visual 3D pipes.

We now present the design of OctoPocus3D that serves as a basis for the experimental studies. Even if multiple 3D gesture guiding systems exist, none of them supports guidance for a large set of hand gestures as required when interacting with a smart environment. Our design approach is to extend OctoPocus to 3D gestures: OctoPocus3D provides revelation and guidance in one step by displaying on-screen 3D pipes.

3. DESIGN ISSUES FOR OCTOPOCUS3D

We first describe the visual presentation of 3D gestures and rationalize its design based on previous studies. This presentation of 3D gestures defines the basis used to then study the impact of different feedback and feedforward alternatives.

3.1 Presenting 3D Gestures

The system represents gestures with 3D pipes and 2D labels displayed next to the pipes. We provide basic feedback with (1) a digital representation of the user's hand as a white sphere and (2), the current performed path in the 3D scene as a white 3D ink trail.

We considered visual cues that help the understanding of a 3D space. The literature suggests about 18 visual cues for space perception, which are application- or task-dependent [27]. It is thus impossible to systematically explore all combinations of the 18 visual cues [27]. For the design of our guiding system, we followed an informal iterative process, which led us to identify the following elements providing depth cues without overloading the 3D scene.

Occlusion happens when an object overlaps another one. It is a strong depth cue that provides binary information: which object is closer / farther from the observer. We hence represent the user's hand with a white solid sphere, so that users can perceive if they are in front or behind the path to follow. However, occlusion of a gesture path by another one is a problem: Users cannot see behind a gesture path that would be in front of others. Thus, as OctoPocus, we applied transparency so that users can still see behind a gesture path and have the depth cue at the same time. Gestures are rendered with two parts: a prefix (first 33% of the complete gesture) and a suffix (last 67% of the complete gesture) more transparent than the prefix (Figure 1).

In order to increase the linear perspective, we added a horizontal textured plane (Figure 1). We chose white stripes on a black background to enhance the contrast of the 3D scene. This plane gives both depth information (parallel lines converging toward the horizon) and viewpoint that the 3D scene is seen from above, with an angle of 45°.

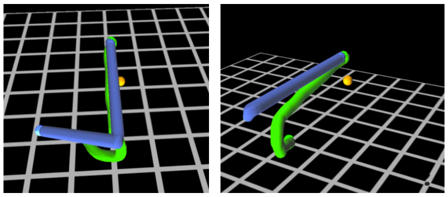


Figure 2: Final feedback with kinetic depth showing intended (blue) and executed (green) gestures from the original perspective (left) and from a tilt to the right perspective (right).

We gave particular attention to the lightning of the scene. While the horizontal plane was too far from the gestures for their shadows to be useful, the shadows created by gestures on gestures themselves provide information regarding corners and curves orientation, called shape-from-shading depth cue [27].

The kinetic depth is a mechanism that provides depth information through motion. Without stereoscopic system, a 3D object will be projected onto the 2D surface of the display. A head-coupled mechanism allows users to rotate the 3D scene, and hence offers the possibility to reveal the 3D shape of the object. To do so, the virtual camera in the 3D scene is set at the position of the user's head, and is pointing toward the center of the scene. This mechanism allows a user to turn around the 3D gestures, i.e. rotate the 3D scene by bending their upper body (Figure 2).

3.2 Feedback Mechanism

With some systems [4,9], gestures less likely to be recognized are thinner than gestures more likely to be recognized. If the prediction is under a threshold, the corresponding gesture is not displayed anymore. Other dynamic guides use the transparency instead [3]. We chose to vary the radius of the gestures' pipes. This visual combination of concurrent feedback and feedforward allows the system to (1) let users correct their gestures if the motor control of arm movements is performed through multiple corrections during the execution of the gesture and (2), conveniently simplify the 3D scene by decreasing radii and making irrelevant gestures disappear.

As highlighted in the related work section, the impact of the concurrent feedback (neither in 2D nor in 3D) on the recognition rate and completion time has not been evaluated on novices yet. It is obvious that this concurrent feedback makes the visual 3D scene simpler during the execution. But do novices really correct their gestures during the execution by using the information provided by the radius of the pipes? We are hence interested in evaluating the impact of the combination of concurrent feedback and feedforward (Figure 1) on the recognition rate and the completion time. We also want to compare this concurrent feedback with the other strategies described in the related work section: no feedback, a terminal feedback, and a combination of concurrent and terminal feedback. Our terminal feedback conveys two types of information (Figure 2). First, the color of the performed path provides binary information: it becomes green (resp. red) if the trial is successful (resp. unsuccessful). Second, users can explore the 3D scene with a head-coupled mechanism in order to get a better understanding of the differences between the intended and the performed gestures displayed on screen.

3.3 Feedforward Mechanism

The feedforward mechanism allows the user to know which commands are available and how to trigger them by showing the associated gestures. With the chosen on-screen visual presentation of gestures with 3D pipes, one of the first challenges is to display several 3D gestures at the same time. Indeed, the third dimension brings an additional difficulty not present in 2D: the depth. The visual complexity depends on several other factors, such as the gestures themselves and the number of gestures in the gesture set. Thus, it is crucial to consider design options that could reduce this visual complexity. From our literature review, we consider two design options that can influence the resulting complexity of the presentation: the quantity of information and the resulting visual stability.

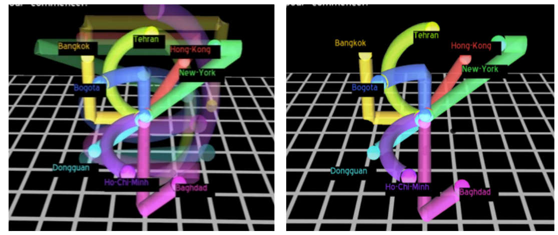


Figure 3: Visualization of the gesture set #3. Left: Remaining portion of gestures. Right: Upcoming portion of gestures.

The first design dimension is the displayed portion of the gestures by the system [4] (Figure 3): The system can present gestures with pipes representing the complete remaining portions of the gestures (like OctoPocus [4]), only upcoming portions of the gestures (like YouMove [2]), or the directions only (like LightGuide [23]). This factor directly impacts the visual complexity of the scene by modifying the quantity of information displayed by the pipes. In our 'upcoming' condition, we chose to represent only the prefix of the gesture (33% of the complete gesture).

A second design parameter is the origin of the guides. For a 2D guiding system, the display of the guide is often collocated with the location of the gesture's execution. Thus, at any time, gesture paths are displayed centered under the point of contact of the interaction tool. However, (1) this design solution might be less accurate than fixed gestures' presentation for 2D gestures [3] and (2), the third dimension adds a new source of noise for positioning the user's hand and causes further difficulty in interpreting the 3D visualization. Indeed, several 3D objects will move and follow the digital representation of the hand, adding instability to the 3D scene. Thus, these two reasons motivated us to evaluate another design alternative: 3D pipes are centered in the scene, and only the digital representation of the hand is moving through the stable representation of the 3D pipes.

For the first experiment dedicated to feedback mechanisms, the feedforward displays the complete remaining portion of the gestures' paths from the center of the 3D scene. This solution has been shown to be better for the case of 2D gestures in [3]. In the second experiment, we then explore feedforward design options considering the best feedback mechanism of experiment 1.

4. EXPERIMENTAL MATERIALS

We describe the gesture sets as well as the apparatus and setup that are common in our controlled experiments.

4.1 Experimental Gesture Set Design

Our goal is to compare different design for presenting the 3D gestures. A desirable goal for a gesture guiding system is to support a rich gesture vocabulary enhancing the expressive power of gestural interaction. Thus, increasing the number of gestures of the set is likely to create additional difficulties. We hence created gesture sets that were deliberately difficult, with self-occlusion (i.e. a gesture part occluding another part of the same gesture), unusual gesture shapes, but also occlusion among gestures (i.e.

gesture behind another gesture), overlapping of gestures (i.e. gestures sharing a portion of their paths) and proximity (both from a spatial and a shape perspective). Finally, we created 4 gesture sets, each of them composed of 8 gestures for a total of 32 gestures. In our experiments, gestures were arbitrarily linked to city names instead of command names. In a real scenario, gestures would be linked to discrete commands such as 'Switch light on'.

4.2 Apparatus and Setup

The two experiments used the same apparatus and setup. The C++ software was developed with Ogre, a 3D rendering framework, and executed on an Intel 2.40GHZ laptop. OpenNi2 and Nite2 libraries allowed for communication with a Kinect 1.0. Gesture recognition used the Dynamic Time Warping implementation of GRT (http://www.nickgillian.com/software/grt). The templates used by the recognizer were synthetic computed 3D path. During the experiments, users were standing two meters in front of the Kinect and screen. For both experiments, since the registration (onset of the gesture) and the termination (end of the gesture) were not the focus of the study, we simply determined them with a click using a mouse held in the left hand.

4.3 Preliminary Study: OctoPocus3D Versus Video Demonstration

First, we wanted to confirm that OctoPocus3D could lead to a better recognition rate than a video demonstration. Indeed a video demonstration is commonly used as a baseline in the literature [2,8–11,13,23] as well as in commercial applications, such as the Apple trackpad preferences. We therefore evaluated OctoPocus3D against a Video Crib Sheet. The version of OctoPocus3D evaluated in the preliminary study displayed the complete remaining portion of the gestures' paths centered in the 3D scene. For each system, a trial was considered successful if the score of the intended gesture was the highest score among the scores of all gestures returned by the recognition algorithm.

For this pilot, we had 4 participants from our laboratory using both systems on two gesture sets. This preliminary study showed that OctoPocus3D led to around 80% of recognition rate (M=80.6%, 95% CI=[67.4%, 93.7%]), and the video crib sheet led to less than 50% of recognition rate (M=46.9%, 95% CI=[29.3%, 64.4%]). The limitation of only 4 participants is considered in the 95% CI. These results confirmed that a guiding system such as OctoPocus3D induces a better recognition rate than videos for such gestures. Indeed, OctoPocus3D has a clear advantage by providing guidance rather than only a visual demonstration. This also confirms that our gesture sets were particularly difficult and that guidance was needed.

5. EXPERIMENT 1: FEEDBACK MECHANISMS

We first explore different types of feedback mechanisms. We analyze their impact on the recognition rate and completion time.

5.1 Participants, Design and Procedure

A total of 20 right-handed subjects participated in the experiment (4 female, 16-47 years-old, M=28.4, SD=7.82). None of them had a background in computer science. Five were accustomed to gestural interaction with video games. The game gestures were learnt with a friend (4 of them) or with a tutorial (1 of them). Thus, none of them had already used a guiding system for 3D gestures.

The experiment lasted approximately 90 minutes per participant. We used a repeated-measure within-subject design. The independent variables were the different types of feedback: concurrent (CONC, yes/no) and terminal (TERM, yes/no) and the repetition number REP (1, 2, 3, 4, 5, 6) in order to analyze the progression over time. The session was composed of 4 blocks: one for each condition, i.e. combination of feedback mechanisms (both, concurrent, terminal and none). Participants used each condition on a different gesture set (4 gesture sets as described above) composed of 8 gestures. The system and the feedback were explained before starting a block. Participants were asked to repeat each gesture 6 times. Gestures were randomly assigned. The order of presentation of the 4 conditions was counterbalanced with a Latin square design. This design resulted in 6 repetitions × 8 gestures × 4 conditions x 20 participants = 3840 acquisitions. At the end of the session, we asked participants to rank systems based on their preferences: from 1 (most preferred system) to 4 (less preferred system).

5.2 Results

The main dependent measures were accuracy, i.e. the recognition rate obtained with a given system, and the duration, i.e. the total trial completion time between the registration step (first mouse click) and the termination step (last mouse click). At any time participants could reset the guide by clicking on the middle-button of the mouse, held in the left hand.

5.2.1 Recognition Rate

We conducted our analysis with three-way repeated-measures ANOVAs. Results show no effect from the concurrent feedback CONC [$F(1.19)=4.27$, $p>0.05$], the terminal feedback TERM [$F(1.19)=3.16$, $p>0.05$] and the repetitions REP [$F(5.95)=1.84$, $p>0.05$] on the recognition rate. Indeed, it appears that as soon as the third repetition, participants manage to address the gap between systems noticeable during the first and second repetition (Figure 4, left). Hence, there is no difference between systems from repetitions three to six. Second, all systems show no or little variation of the recognition rate over time. Thus, showing concurrent or terminal feedback did not help participants to progress as expected.

We further refine our analysis by focusing on the two first repetitions, i.e. the very first use of the system. There is a significant effect of CONC on the recognition rate [$F(1.19)=8.20$, $p<0.01$, $\eta^2=0.05$]: Concurrent feedback leads to 10% better recognition rate than no concurrent feedback (Figure 4, right). On the contrary, TERM does not have a significant effect on the recognition rate during the first two repetitions [$F(1.19)=1.17$, $p>0.05$].

From these results, we draw two conclusions. First, concurrent feedback allows a better recognition rate during the familiarization with the guiding system. This is likely due to the influence of the concurrent feedback on the representation of the gestures: indeed within the time interval of a gesture, the concurrent feedback rapidly simplifies the 3D scene. Second,

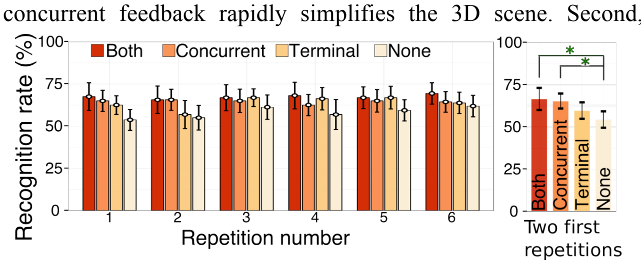


Figure 4: Left: Recognition rate (%) for each combination of feedback across all six repetitions. Right: Recognition rate (%) for each combination of feedback during the two first repetitions. Error bars represent 95% confidence intervals.

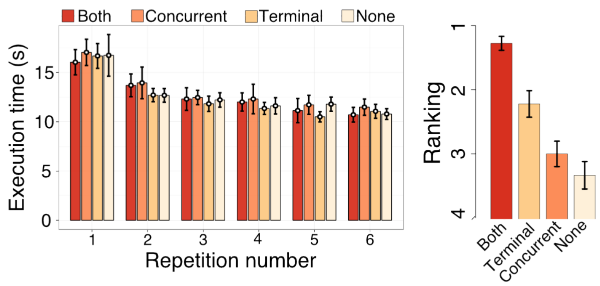


Figure 5: Left: Completion time (seconds) for each combination of feedback across all six repetitions. Right: Qualitative ranking of each combination of feedback (preferred system is ranked 1st). Error bars represent 95% confidence intervals.

participants exhibited a rather fast adaptation to all systems, revealing that they are equally accurate after only two repetitions whatever the feedback provided by OctoPocus3D.

5.2.2 Completion Time

A three-way repeated-measures ANOVA shows a significant effect from the repetition on the completion time [$F(5.95)=35.06$, $p<0.001$, $\eta^2=0.18$]. Post-hoc multiple pairwise t-test comparisons with Bonferroni correction show a constant significant acceleration until the third repetition [$p<0.05$] (Figure 5, left). Three repetitions seem to be enough for the completion time to stabilize. Indeed, although the 6th repetition is significantly different from the 3rd one [$p<0.001$], repetitions number 5 and 6 are not significantly different [$p>0.05$].

Interestingly, we did not find any significant effect from CONC [$F(1.19)=1.15$, $p>0.05$] and TERM [$F(1.19)=0.55$, $p>0.05$] on the completion time. This is surprising as we expected participants to slow down with CONC in order to correct their gestures during the interaction by considering the radius of the pipes. On the contrary, most participants reported that they did not exploit the concurrent feedback. This assertion is supported by their rankings: the two systems providing terminal feedback (both and terminal) were preferred [$\chi^2(3)=28.48$, $p<0.001$] (Figure 5, right). We suppose that even if participants did not take the time to correct their gestures, the concurrent feedback has the advantage of reducing the visual complexity during the execution of the gesture, and hence easing the first two executions of gestures.

6. EXPERIMENT 2: FEEDFORWARD MECHANISMS

We now focus on the two feedforward dimensions: the quantity of information delivered by the system (i.e. displayed portion of gesture's paths) and its visual stability (i.e. origin of the guides).

6.1 Participants, Design and Procedure

A total of 24 right-handed subjects participated in the experiment (10 female, 18-43 years-old, M=25.0, SD=7.7). As for experiment 1, none of them had a background in computer science. 14 of them were used to gestural interaction with video games. Amongst them, 5 learnt the gestures through video demonstration or pictorial images, 5 through tutorial and 4 with demonstrations by a friend. Thus, none of them had already used a guiding system for 3D gestures.

The experiment lasted approximately one hour per participant. We used a repeated-measure within-participant design. Since feedforward mechanisms are important during the very first uses of the system, we wanted to evaluate how these design options affect recognition rates *before* users learn the gesture. Thus, we chose a design preventing progress over time. Participants were guided by a system only once on a given gesture set before using another system with a different gesture set. They repeated the overall process 3 times. Before each use of a system, users were verbally reminded of its characteristics. Gestures were randomly assigned. The order of presentation of the systems was counterbalanced with a Latin square design. This design resulted in 8 gestures × 4 system × 3 repetitions x 24 participants = 2304 acquisitions. The independent variables were the displayed portion of a gesture DISP (portion, complete) and the origin of the guide ORI (center, hand). The 'portion' condition displays only the 33% upcoming part of the gesture while the 'complete' condition displays the complete remaining part of the gesture. The 'center' condition displays the guides centered in the 3D scene, while the 'hand' conditions displays the guides gathered around the digital representation of the user's hand. At the end of the session, we asked participants to rank systems based on their preferences: from 1 (most preferred system) to 4 (least preferred system).

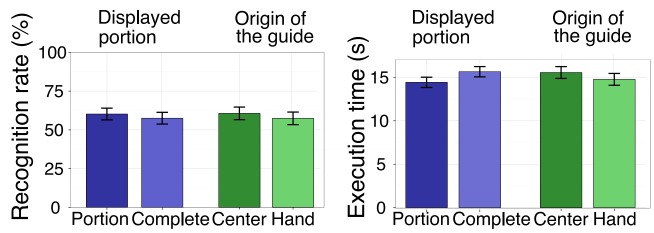


Figure 6: Left: Recognition rate (%). Right: Execution time (s). Error bars represent 95% confidence intervals.

6.2 Results

As for experiment 1, the main dependent measures were the recognition rate and the completion time.

6.2.1 Recognition Rate

A two-way repeated-measures ANOVA reveals no effect from the displayed portion [$F(1.23)=1.09$, $p>0.05$] or the origin of the guide [$F(1.23)=1.34$, $p>0.05$] on the recognition rate (Figure 6, left).

We did not expect these results since these alternatives greatly influence the visual complexity of the resulting scene. Previous work often considers the visual complexity as an important factor for a guiding system [2,4,22]. However, we did not find any significant effect regarding the recognition rate when using OctoPocus3D.

6.2.2 Completion Time

A two-way repeated-measures ANOVA reveals a significant effect from the displayed portion on the completion time [$F(1.23)=9.14$, $p<0.01$, $\eta^2=0.01$], displaying an upcoming portion of the gesture leading to 8% faster completion time that displaying the complete remaining portion (Figure 6, right). This shows that for rapidity, the overall simplicity of the scene overrides the benefit of the anticipation allowed by the display of the entire remaining gesture. We did not find any significant effect from the origin of the guide on the completion time [$F(1.23)=2.86$, $p>0.05$].

Interestingly, feedforward design options also did not have a significant effect on the participant's preferences [$\chi^2(3)=1.38$, $p>0.05$].

7. Discussion and Limitations

Results allow us to draw design recommendations for distant on-screen gesture guidance.

First, regarding feedback alternatives, if the system knows which gesture the user wants to perform (e.g., practice phase), the guiding system should include both concurrent (better recognition rates during the first trials) *and* terminal feedback (users' preferences). However, if there is no practice phase enabling terminal feedback, then using only concurrent feedback is a viable solution. In addition, we showed that this concurrent feedback is beneficial only during the first two repetitions using OctoPocus3D. This result is important since it experimentally reinforces the recommendation for early removal of concurrent feedback: indeed previous results showed that the removal of the concurrent feedback allows avoiding of negative impact on motor learning when users become experts [3]. We additionally show that the concurrent feedback does not contribute to better performances after the first two repetitions. Our design recommendation is then to provide concurrent feedback for novice users only, as well as terminal feedback if possible.

Second, regarding feedforward alternatives, since

1. we did not find a significant effect on accuracy,
2. there is no significant difference in users' preferences [$\chi^2(3)=1.38$, $p>0.05$], and
3. displaying the upcoming portions of gestures led to a 8% faster completion time than displaying the complete remaining portions,

our design recommendation is based on the completion time. While previous work shows no significant impact of the guide on completion time in 2D [3] and 3D [20], we recommend that the guidance system displays only upcoming portions of gestures.

These two design recommendations fall within the broader topic of motor control. Indeed, models for the motor control of arm movements are actively discussed amongst researchers [7]. Arm movements can be planned (1) prior to the actual movement (open-loop control), (2) during the actual movement through multiple corrections (closed-loop feedback control) or (3), both before the movement and modified during the actual movement (a mix of closed- and open-loop control). Our design recommendations can be explained by the closed-loop feedback control of the arm movements. This means that the users constantly update their arm movements according to what they perceive of the situation. Indeed, concurrent feedback and the display of only upcoming portions of gestures advocate for a continuous correction process. Contrastingly, a terminal feedback and the display of the complete remaining portions of the gestures would have helped for pre-programmed movements, performed under open-loop control.

The visual scene instability induced by the origin of the guides should interfere with the closed-loop feedback control of the arm movements. Nevertheless we did not experimentally observe this effect. We hypothesize that even if users executed their movements under closed-loop feedback control, taking benefit from the visual information, only small visual portions of the upcoming paths were considered. For this case, the focus of the users is mainly on the digital hand representation and nearby displayed portions of the gestures. This area is stable in both conditions (guides with fixed origin or guides following the digital hand), even with a globally unstable 3D scene. This hypothesis should be validated through formal experiments by monitoring user's gaze while being guided. Tracking the eyes of the users while being guided could (1) validate this hypothesis and (2), define the useful area around the digital hand for presenting 3D gestures. This result could then help minimizing the visual complexity of the 3D scene by optimizing the displayed portions of gestures.

For this study, we used *a priori* difficult gesture sets in order to find differences in a difficult case to support richer gesture sets. Easy gesture sets – such as a gesture set composed of only two opposite straight lines – would have likely hindered potential effects of the explored design options. However, it is difficult to clearly define and quantify the difficulty of a gesture. Indeed, there are several factors to consider. A gesture can be (1) perceived difficult by users [24], (2) difficult to execute because of its geometrical factors such as its shape, and/or (3), difficult to recognize because of similar gestures in the gesture set. The first step toward our understanding of gesture difficulty (perceived, motor execution and recognition for interaction) is the establishment of a guiding system. Indeed, sources of difficulty will likely be related to the way gestures are presented to users. For instance, a horizontal line will be more difficult to execute if it is seen from a horizontal viewpoint than with a vertical viewpoint from the top. But what happens if another gesture from the gesture set occludes the line? Thus, the way gestures are presented to users is a critical aspect for the study of gesture difficulties. In this study, we introduced a guiding system that is required in order to perform further studies in which users can actually execute a 3D semaphoric dynamic gesture from a gesture set. A next step is thus the analysis of the difficulties of the gestures as previously mentioned: perceived, motor execution and recognition difficulty of a gesture from a gesture set.

8. Conclusion

3D gestures are becoming widespread in applications such as games, public displays and smart environments. However, such a gestural interaction is not self-evident: users have to learn the gestures corresponding to the system commands. In addition, users might struggle with the functioning of the system and how it processes gesture inputs. Building on previous results regarding 2D gesture sets guidance and 3D gestures guidance, this paper addresses the design of an extension of OctoPocus in 3D, i.e. a guide displaying gestures as on-screen 3D pipes. Previous literature offers several combinations of feedback and feedforward mechanisms, but lacks an evaluation of their impact on the interaction. As a consequence, it is difficult to inform design decisions for display of 3D gestures. We presented two studies that enrich the knowledge of 3D gesture guidance and allow the fine-tuning of the representation of 3D gestures on a distant screen.

Our contributions are twofold:

Firstly, we extend OctoPocus to 3D semaphoric dynamic gestures with OctoPocus3D. We show that this straightforward extension provides a better guidance than video demonstrations. We thus encourage researchers and practitioners to use this extension since its gesture sets are fully flexible: i.e., gestures can be easily re-defined, unlike videos that need to be re-shot.

Secondly, results from two users experiments allow us to draw design recommendations regarding the fine-tuning of OctoPocus3D. As regards to feedback options, we showed that concurrent feedback visually simplifies the 3D scene and, during the first two executions of a gesture, leads to a 10% higher recognition rate than no feedback. A previous study [3] showed that concurrent feedback should be removed in order to enhance learning; we further show that concurrent feedback is useful, but only at the very beginning. The resulting guiding system, OctoPocus3D, can be used without any terminal feedback, i.e. without any training session during which terminal feedback allows the user to compare the intended and executed gestures. As regards to feedforward options, we showed that displaying only a portion of gestures with 3D pipes allowed a 8% faster completion time than displaying the entire gesture paths. Indeed, even if this option prevents users from anticipating the complete shape of the gestures, this option also reduces the visual complexity, a salient property for the case of 3D gestures.

We plan to extend this study along two complementary avenues, the first one dedicated to the research question raised by our results regarding user's focus and the second one related to the study of the gesture difficulty.

First, we plan to further explore the visual focus of the user during guidance using OctoPocus3D. This could provide a better understanding of how users execute arm movements during guidance under closed-loop feedback control and also increase the legibility of the 3D scene by optimizing the displayed portions of the gestures.

Second, we plan to study the difficulties of 3D semaphoric dynamic hand gestures. More specifically, we want to study the difficulty of motor execution (i.e. what makes a gesture difficult to execute?) and how this difficulty is linked to the viewpoint of the guide.

9. ACKNOWLEDGMENTS

This work has been supported by the DELight project (French government's FUI -Single Inter-Ministry Fund- program) and by ANR (PERSYVAL-Lab grant ANR-11-LABX-0025-01 and AP2 project ANR-15-CE23-0001-02).

10. REFERENCES

[1] Aigner, R. et al. 2012. *Understanding Mid-Air Hand Gestures : A Study of Human Preferences in Usage of Gesture Types for HCI*. MSR-TR-2012-111.

[2] Anderson, F. et al. 2013. YouMove: Enhancing Movement Training with an Augmented Reality Mirror. *Proc. UIST '13*, ACM Press, 311–320. DOI= http://dx.doi.org/10.1145/2501988.2502045

[3] Anderson, F. and Bischof, W.F. 2013. Learning and performance with gesture guides. *Proc. CHI '13*, ACM Press, 1109–1118. DOI= http://dx.doi.org/10.1145/2470654.2466143

[4] Bau, O. and Mackay, W.E. 2008. OctoPocus: A Dynamic Guide for Learning Gesture-Based Command Sets. *Proc. UIST '08*, ACM Press, 37–46. DOI= http://dx.doi.org/10.1145/1449715.1449724

[5] Bragdon, A. et al. 2009. GestureBar: Improving the Approachability of Gesture-based Interfaces. *Proc. CHI '09*, ACM Press, 2269–2278. DOI= http://dx.doi.org/10.1145/1518701.1519050

[6] Delamare, W. et al. 2015. Designing guiding systems for gesture-based interaction. *Proc. EICS '15*, ACM Press, 44–53. DOI=http://dx.doi.org/10.1145/2774225.2774847

[7] Desmurget, M. and Grafton, S. 2000. Forward modeling allows feedback control for fast reaching movements. *Trends in Cognitive Sciences*. 4, 11 (2000), 423–431. DOI= http://dx.doi.org/10.1016/S1364-6613(00)01537-0

[8] Fothergill, S. et al. 2012. Instructing people for training gestural interactive systems. *Proc. CHI '12*, ACM Press, 1737–1746. DOI= http://dx.doi.org/10.1145/2207676.2208303

[9] Freeman, D. et al. 2009. ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. *Proc. ITS '09*, ACM Press, 165–172. DOI= http://dx.doi.org/10.1145/1731903.1731935

[10] Hoffman, M. et al. 2010. Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices. *Proc. VR '10*, IEEE, 59–66. DOI= http://dx.doi.org/10.1109/VR.2010.5444813

[11] Kamal, A. et al. 2014. Teaching motion gestures via recognizer feedback. *Proc. IUI '14*, ACM Press, 73–82. DOI= http://dx.doi.org/10.1145/2557500.2557521

[12] Karam, M. and Schraefel, m. c. 2005. A Taxonomy of Gestures in Human Computer Interactions. (2005), 1–45.

[13] Kühnel, C. et al. 2011. Im home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human Computer Studies*. 69, 11, 693–704. DOI= http://dx.doi.org/10.1016/j.ijhcs.2011.04.005

[14] Kurtenbach, G. et al. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction 8*, 1–23. DOI= http://dx.doi.org/10.1207/s15327051hci0801_1

[15] Kurtenbach, G. et al. 1994. Contextual Animation of Gestural Commands. *Computer Graphics Forum*. 13, 5, 305–314. DOI=http://dx.doi.org/10.1111/1467-8659.1350305

[16] Nacenta, M. a et al. 2013. Memorability of pre-designed and user-defined gesture sets. *Proc. CHI '13*, ACM Press, 1099–1108. DOI=http://dx.doi.org/10.1145/2470654.2466142

[17] Norman, D.A. 2010. The way I see it: Natural user interfaces are not natural. *interactions*. 17, 3, 6–10. DOI= http://dx.doi.org/10.1145/1744161.1744163

[18] Portillo-Rodriguez, O. et al. 2008. Real-Time Gesture Recognition, Evaluation and Feed-Forward Correction of a Multimodal Tai-Chi Platform. *Haptic and Audio Interaction Design*. Springer. 30–39. DOI= http://dx.doi.org/10.1007/978-3-540-87883-4_4

[19] Ren, G. and O'Neill, E. 2012. 3D Marking menu selection with freehand gestures. *Proc. 3DUI '12*, IEEE, 61–68. DOI= http://dx.doi.org/10.1109/3DUI.2012.6184185

[20] Rovelo, G. et al. 2015. Gestu-Wan - An Intelligible Mid-Air Gesture Guidance System for Walk-up-and-Use Displays. *Proc. Interact '15*, Springer. 368–386. DOI= http://dx.doi.org/10.1007/978-3-319-22668-2_28

[21] Schmidt, R. a. and Wulf, G. 1997. Continuous Concurrent Feedback Degrades Skill Learning: Implications for Training and Simulation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*. 39, 4, 509–525. DOI= http://dx.doi.org/10.1518/001872097778667979

[22] Schönauer, C. et al. 2012. Multimodal Motion Guidance: Techniques for Adaptive and Dynamic Feedback. *Proc. ICMI '12*, ACM Press, 133–140. DOI= http://dx.doi.org/10.1145/2388676.2388706

[23] Sodhi, R. et al. 2012. LightGuide: Projected Visualizations for Hand Movement Guidance. *Proc. CHI '12*, ACM Press, 179–188. DOI=http://dx.doi.org/10.1145/2207676.2207702

[24] Vatavu, R.-D. et al. 2011. Estimating the Perceived Difficulty of Pen Gestures. *Proc. INTERACT'11*. Springer. 89–106. DOI=http://dx.doi.org/10.1007/978-3-642-23771-3_9

[25] Vatavu, R.D. 2013. A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *Journal of Ambient Intelligence and Smart Environments*. 5, 2, 187–211.

[26] Vatavu, R.D. 2012. Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *Journal of Ambient Intelligence and Smart Environments*. 4, 2, 79–93.

[27] Ware, C. 2004. *Information visualization: perception for design*. Elsevier.