Visual Gesture Mapping

Kasper Rise and Ole Andreas Alsos

[1] Norwegian university of Science and Technology, Norway
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
`lncs@springer.com`

Abstract. The abstract should summarize the contents of the paper in short terms, i.e. 150-250 words.

Keywords: Gesture-based Interaction, virtual reality, Gesture control.

1 Introduction

Keyboards, touch screens, buttons, levers and joysticks are common ways of controlling and interacting with computers, machines and robots. In most cases these conventional ways of interacting with computers are sufficient, but sometimes they restrict our interaction. Gesture control, defined as the ability to recognize and interpret movements of the human body in order to interact with and control a computer system without direct physical contact, removes the input controller device and allows the user to interact directly with the machine, using their hands or body ("Gesture Control," 2019). Gesture-control can, therefore, offer a more intuitive and natural interaction than conventional interaction.

In robotics, there are several advantages of implementing gesture-based interaction (Rise, 2020). Firstly, replacing advanced controllers with simple and intuitive gestures can lead to a clear improvement of intuitiveness and reduce the time it takes to learn the system. Additionally, controlling robotics with discrete hand gestures can be highly beneficial in environments where the body is physically restrained. The challenge is that there are no agreed-upon interaction patterns for gesture-based interaction in VR and AR environments. Therefore, we have in this study developed a set of 10 gestures that can solve many common interaction patterns in domains where gestures can be used to interact with systems. We hope that these gestures can be a starting point for a standard convention for gesture-based interaction in a real, virtual (VR) and augmented reality (AR).

The purpose of the study was to:

1. develop a set of intuitive gestures that are simple, logically mapped and non-strenuous
2. design corresponding visualization-feedback based on design principles

3. evaluate if visualization feedback helps the user learn and understand the gestures

We have limited our focus to *active* gestures, i.e. gestures giving continuous feedback and not gestures that are a part of a «gesture language» where one gesture, or combination of gestures in sequence, is mapped to one system command.

We have used an iterative design approach in developing the gestures, using a range of methods such as literature review, co-design with experts on gesture-based interaction of drones, usability testing, surveys and post-test interviews with novices.

In the rest of the article, we present further details on the theoretical foundation and background, describe the methods used, describe the 10 gestures developed, present data from one of the usability tests and discuss the findings from the test.

2 Background

2.1 Gesture-based Interaction

The term *Gesture contro*l is used in a wide range of contexts and can be interpreted differently. In Gartner Glossary gesture control is defined as "(...) the ability to recognize and interpret movements of the human body in order to interact with and control a computer system without direct physical contact" ("Gesture Control," 2019). Although this definition covers any movements of the human body, we will in this article focus on hand gestures.

Applications of gesture-based interaction. Even though gesture-based interaction is a relatively new interaction method, it has emerged in several domains already. TeMoto, a teleoperation system has been developed, using a Leap Motion sensor to track gestures to control a rover (Valner et al., 2018). Gestix uses vision-based gesture tracking for navigation and manipulation of electronic medical records (EMR) (Wachs et al., 2008). Using gesture control to operate a robotic microscope in surgery (Antoni, Sonnenburg, Saathoff, & Schlaefer, 2015), laparoscopic instruments (Arkenbout, de Winter, Ali, Dankelman, & Breedveld, 2018) and for controlling operating light (Hartmann & Schlaefer, 2013) have also been explored in the healthcare domain. Research has been done using sensor gloves to control a snake-like robot in space (Liu, Luo, & Ju, 2016) and in the field of military technology a so-called "real-time soldier-robot teaming" have been developed using multimodal interaction, combining gesture control and speech (Barber, Howard, & Walter, 2016). We also see gesture-based interaction emerge in the consumer market, such as in premium smartphones (Google's Pixel 4) and cars (BMW).

2.2 Design Theory

Gesture-based interaction in a real or VR/AR environment is not different from other forms of interaction; it has to follow the basic rules of interaction design. This means that *visibility*, *feedback*, *constraints*, *mapping* and *affordance* need to be considered when designing these. In this section, Norman's design principles (Norman, 2010) will be explained and put in the context of gesture-based interaction.

Visibility and Feedback. Gestures-based interaction is not a conventional interaction pattern that everybody are familiar with. Therefore, if you are using it for the first time, you would need the proper visual clues to aid you in knowing what to do and how to do it right. In a real environment, it can be challenging to provide this feedback, especially when operating IoT devices, where the user interface often is very limited or only accessible through a smartphone or computer. In augmented reality (AR) or virtual reality (VR), it is easier to provide visual interaction feedback inside the virtual model.

There are several ways of giving interaction feedback. The feedback can be sound, light, vibrations, screen-based visuals or text, etc. AR and VR both provide gesture-based interaction with the visual feedback that is needed to obtain an intuitive interaction. In this study, we have developed visualization that provides the necessary visual feedback to the users.

Dynamic feedback is one type of feedback that stands out as being highly valuable in the context of gesture-based interaction scenario. Dynamic feedback means giving constant feedback based on the movement of the gestures, in contrast to summative feedback given only when an interaction is performed and successfully interpreted by the system. In a VR environment where physical hand controls are used, this is often displayed as a pair of virtual hands floating in mid-air showing which buttons are pressed or not (ref). By using gesture tracking, one can get constant input from the users' gestures, and give feedback accordingly to what they are doing. This increases the intuitiveness of the interaction. The goal is to give enough feedback to allow the user to interact with the system without the need for instructions.

Constraints. When designing physical or digital objects, giving physical or visual constraints one can give the user the hints needed to intuitively understand how to (and not to) interact with the object. Real-world examples are garbage bins with different openings depending on the categorization of the trash is a common example of constraints that forces the user to throw the trash in the correct bin (Norman, 2010).

When designing for gestures one has to consider biomechanics and ergonomics. Bodily constraints, such as how you rotate your arm, fingers and wrist, are important to take into consideration when designing for physical controls, such as keyboards, joysticks and throttle controls. Constraints can also be used as an aid when designing visualizations for a VR/AR environment. By emphasizing biomechanics and the bodily constraint in the visualization, the user will intuitively understand the boundaries of the gestures, knowing what is maximum and minimum. An example from the physical world is the power throttle of a boat. Based on its physical restrictions the user knows when the throttle is on max because the throttle can not move any further forward.

Mapping. Natural mapping is when there is a clear and obvious relationship between the controls and the object to be controlled. The gesture being used to determine an action should be mapped correspondingly to the action being executed. In a study on gesture control in VR they discovered that:

> The gesture/action should be well mapped to the content being learned. An example of this could be mapping changing gears in a car. The action should be similar to moving the gear stick rather than pushing a toggle up or down. By performing actions, it stimulates the motor system and also strengthens the memory traces associated with newly learned concepts (Johnson-Glenberg, 2018).

Every visualization should, therefore, be mapped according to the position of the hand, considering physical restrictions as well. Intuitive metaphors should also be used both when choosing the gesture and the visualization. An example could be an axial plane describing the movement of a drone.

Affordance. Affordance refers to an attribute of an object that allows people to know how to use it. Essentially, *to afford* means to give a clue. As you intuitively know how to open a door by looking at where its hinges and doorknob are located, affordances can be used to design visualizations for gestures that work as a «hint» of how it should be used. This could be translated into direct visual elements that indicate something or when choosing visual elements already known. An example could be choosing known geometry that has the affordances of what you want to convey. A cylinder has the affordance of being rotated along its axis and a sphere or a ball has the affordance of scaling about origo (Norman, 2010).

2.3 Guidelines

Combined with design theory, having certain guidelines as a framework when designing for gestures, is beneficial. The guidelines in this chapter is based on literature research, taking knowledge from different research-domains. Some of the guidelines are also requirements from relevant stakeholders that work as a framework.

Gesture as assessment. In a study using gesture control in VR for education, Johnson-Glenberg states that designing for gestures should reveal the state of the learner's mental model, both during learning (called formative or in-process) and after the act of learning (called summative). She gives the example;

> Prompt the learner to demonstrate negative acceleration with the swipe of a hand controller. Does the controller speed up or slow down over time? Can the

> learner match certain target rates? This is an embodied method to assess comprehension that includes the added benefit of reducing guess rates (Johnson-Glenberg, 2018, p. 16).

Although this literature is mainly focusing on gestures for learning, this idea can be applied to any case of designing for gestures. In our case, the visualizations should give a clue what the user is supposed to do before doing an action, meanwhile doing the action (formative) and give a response after an action has been done (summative). These three steps should be covered with constraints, affordances, feedback and good mapping as discussed in the previous section.

One theory that further underlines this concept is that using gestures requires motor planning and this activates multiple simulations even before the action is taken. Hostetter and Alibali (2008) posit that:

> Gestures first requires a mental simulation before movement commences, at that time motor and premotor areas of the brain are activated in action-appropriate ways. This pre-action, covert state of imaging an action appears to stimulate the same collaries as the overt action i.e., motor cortex, the cerebellum, and basal ganglia (Jeannerod, 2001). The combination of planning and then performing may lead to more motor and pre-motor activity during encoding, which might lead to a stronger learning signal and memory trace.

Enabling user-defined gestures. In a paper that discusses the ability of users to remember the gestures and their meanings (Nacenta, Kamber, Qiang, & Kristensson, 2013), the authors considered user-defined gestures, pre-designed gestures and randomly assigned gestures and concluded that user-defined gestures are preferable and more memorable than the other types. The authors recommended 'enabling user-defined gestures' as well as paying attention to the relationship between the gesture and the action that it invokes. This coheres with other recent work on natural user interfaces(NUIs)((Malizia & Bellucci, 2012); (O'hara et al., 2013)).

This theory is the core concept of designing a library of different gestures. By making a library instead of imposing gestures where they are unnatural or might not fit the purpose, users can choose the gestures and corresponding visualizations that fit the use case.

3 Developing a Gesture Library

In this section, we present the process and results of developing a gesture library and corresponding visualizations, which could be used as interaction feedback in a AR and VR setting.

3.1 Method

The design theory and literature review presented above worked as a framework when selecting gestures and design the gesture visualizations presented later in this paper. Even though the gestures were chosen rather than "designed", they had to follow the previously established guidelines and correspond with the visualization. It was, therefore, two elements that had to be designed together. The first step was to create a framework of criteria that was crucial for the success of the project. The second step was to understand what kind of gestures different use-cases would need. A co-design workshop with designers, drone pilots, VR-experts and computer engineers was conducted focusing on establishing categories of gestures that could be used in an real or AR/VR setting. The criteria were established based on prior experience working on the topic of gesture control of drones, requirements from stakeholders and take-aways from use-cases. The process then continued with mapping out several use cases, creating user-journeys to understand which type of interaction that would best suit the different case.

Categories of gestures. From the workshop, we came up with 3 main categories: (1) gestures for *directional movement*, e.g. steering a rover, (2) gestures for *flow control*, e.g. controlling the speed of a rover (3) gestures for *spatial orientation,* e.g positioning a robotic arm. While these categories were sufficient to cover simple (meaning single output) gestures, they did not cover combinations or variations of gestures. Therefore, two subcategories were developed: (4) *multifunctional* gestures, e.g. controlling the path *and* speed of a rover, and (5) *tactile* gestures, e.g. selecting the power-level or gears.

When the categorization was established, the process of designing spesific gestures for each category started. Using the use cases that were discovered in the workshop and the background from research and design theory, a set of gestures were designed for each of the three main categories. Then these gestures were used to implement new gestures for the use-cased that required more than one control output or would benefit from more tactile feedback.

When a gesture library was established, they were reviewed in a second co-design workshop with the same experts on VR and gesture interaction with competence in both interaction design, computer science and artificial intelligence. The group gave feedback on design and interaction, technical implementation, input algorithms and possible use scenarios. With the feedback and new scenarios, some gestures were iterated, some discarded and new ones were designed. Finally, a gesture library comprising of 10 gesture visualizations were designed. One gesture was chosen for further development and testing on the background of being one of the most valuable as well as relatively simple to implement and test in a VR environment. Details on the user test presented in the section *Usability Evaluation*.

3.2 Findings

Criteria for designing gestures. Based on the scope of the project and the use case, certain criteria can be formulated to narrow down the research area. The criteria, compared to the guidelines, are not based on literature or research, but rather on input from experts.

The gesture should be unique and non-overlapping. The participants of the co-design workshops reported that when controlling robotics, several inputs are required. Designing gestures that are unique would, therefore, make it possible to control several mechanics with a few distinct gestures. The gestures should, in that case, be designed in a way that they do not overlap to avoid them being interpreted as two things at once. One example the participants used was controlling a drone using your palm, where the power is adjusted by bending your fingers. To activate the automatic landing function of the drone, one has to make a fist. In this case, when making a fist to land, the input will firstly be to increase the power, making the drone go up, before registration the fist gesture making it activate the landing function.

Gestures should require as little movement as possible. When designing for interaction using hand gestures, one has to consider biomechanics and the physical strain on the body. As a guideline, one should therefore always try to avoid bodily movements that are physically demanding or that will cause strain or fatigue over time. An argument for using gestures to control robotics could be that it is used in scenarios where the human body is physically constrained, thereby using facile gestures is beneficial.

Categorization of Gestures. The workshops with the experts on gesture-based drone control resulted in 5 different gesture categories for interacting with various objects. These categories were (1) gestures for directional movement, (2) gestures for flow control, (3) gestures for spatial orientation, (4) multifunctional gestures, and (5) tactile gestures. A description of each category is presented in table 1.

Table 1. Description of each gesture-category

Main Categories	Description
1 Gestures for directional movement	Gestures related to directional movement and does not deal with the control of altitude etc.
2 Gestures for flow control	Gestures related to controlling the amount of any output value

3 Gestures for spatial orientation	Gestures related to three-dimensional orientation and movement in 3 degrees of freedom (DOF) along the x,y and z-axis
Sub Categories	
4 Multifunctional gestures	Gestures controlling more than one input at the time and therefore combine gestures from the other categories
5 Tactile gestures	Gestures using the body itself as tactile feedback. These gestures are not unique on their own but can be merged with the other gestures in cases where tactile feedback is required

Developing a gesture-library. In addition to the gesture categories, we developed two gestures in each of the five categories, giving a total of 10 gestures. An overview of the gestures is presented in Figure 1. A detailed description of the gesture categories and gestures is presented in the following subsections.

Fig. 1. Ten gestures and corresponding visualizations.

Gestures for Directional Movement. Controlling the directional movement is related to any steering mechanism. Mechanical steering is often more constrained than flow control and spatial orientation, and should, therefore, be used with gestures that follow these constraints.

1. *Pointing Arrow.* In use cases where a more directional (rather than rotational) visualization is more intuitive can the pointing gesture be applied. Making a flat hand with straight fingers, moving around the wrist, one can point in the desired direction. The gesture is constrained to a certain degree which is also added in the visualization.

Fig. 2. Gesture 1: *Pointing Arrow* gesture and visualization

The visualization consists of an arrow pointing in the direction of the gesture. The arrow has the affordance of direction and the circle at its root indicates rotation. Adding a stapled line in the center makes it easy to indicate where the neutral position is.

2. Rotating Cylinder. The gesture of rotating your wrist around the axis of your hand gives you close to 180 degrees of rotational freedom. This matches the mechanical constraints of steering. In this case, the hand gesture, either it is a palm or fist, is not that important, which gives it the benefit of being combined with other gestures (which will be further explored in Multifunctional Gestures section).

Fig. 3. Gesture 2: *Rotating Cylinder* gesture and visualization

The cylinder has the affordance of rotation. To give the cylinder an index of the rotational position, a vertical plane is added. This plane represents the position of the hand, where the «flatness» of the hand is mapped to the plane. The cylinder itself is mapped in the direction of the arm, making it an abstract illustration of the user's own arm. To make the physical constraints cohere with the visualization, two balls are added to stop the rotation where the hand can not be rotated anymore.

Gestures for Flow Control. Controlling the flow could be any control unit measuring an amount from 0 to a limit (i.e power, volume, throttle, etc). Simple gestures that represent this control unit can be used either alone, combined with two hand gestures(another gesture for the other arm), or combined with other controls in the same gesture(see Multifunctional gestures). By making the visualization in 3D it gives a better representation as it can be better mapped to the direction of the gesture.

3. Cylinder Flow Bar. The gesture of extending and closing the fingers follows the same constraints as the visualization. A fully stretched out hand can not be extended anymore, making it a natural max point. A closed palm, therefore, represents the minimum point as the gesture can not contract any further. The gesture is also ergonomic and non-straining and has a sufficient span to give control over fine-adjustment.

Fig. 4. Gesture 3: *Cylinder Flow Bar* gesture and visualization

The cylinder has the clear constraints by having a defined height/length, determining the max and min. It also has clear visual dissemination of filling up a cylindrical container and works as a metaphor for filling up a glass. Being in 3D, the visualization can be mapped to the direction of the gesture, making the dynamic feedback even more intuitive.

4. Ball in Sphere. The gesture of opening and closing the fist is similar to the previous gesture, but represent a ball rather than a cylinder. The gesture also has the constraints of being at the min with a closed fist and at max with a fully extended hand. The gesture is also ergonomic and non-straining.

Fig. 5. Gesture 4: *Ball in Sphere* gesture and visualization

The visualization of a ball in the sphere follows the hand being mapped as a ball when being a fist. As the user extend the fist/ball gesture the ball will scale up until it reaches its map being when the hand is fully extended(but still mimicking a ball). The sphere around the ball represents the max.

Gestures for Spatial Orientation. Spatial orientation, meaning positional orientation in a three-dimensional space, is harder to design for, as there are less common metaphors to represent the controls. One could argue that spatial orientation is harder than previous mentioned orientations and a well-designed visualization can, therefore, provide even more value.

5. Plane gesture. Some control units do not have limited directional navigation but can have 360-degree navigation. A plane is a good metaphor to represent navigation on one level. A flat hand can then be used to represent the plane resulting in a good mapping. To give feedback on the power being applied in the direction, a grid system can be used to visualize how much power and in what direction the power is applied. By moving the hand like a plane the visualization will correspond showing the corresponding visualization.

Fig. 6. Gesture 5: *Plane gesture* and visualization

The visualization is designed as a grid system, where the power and direction are indicated by coloring the squares according to the gesture. The borders of the grid work as a constraint indicating the maximum of power. The black square in the middle indicates the neutral position.

6. XYZ gesture. While the gesture plane gesture gives 360 degrees of navigation, it only covers the x and y-axis. In situations where you want to position or control something in all directions(along the x,y and z-axis) the XYZ gesture can be valuable. The distinct gesture of simulating the XYZ axis with your fingers makes it intuitive and well mapped.

Fig. 7. Gesture 6: *XYZ gesture* and visualization

The visualization includes four elements; a ball, a cylinder, an XYZ axis and a sphere. The ball represents the hand's position in the real space. The cylinder visualizes the power being applied as its thickness increases as it gets further away from origin (the point set when activating the gesture). The XYZ axis is there as a visual aid to show which direction and position in space the force is being applied. And the sphere is there as a constraint to visualize the maximum of power one is able to apply.

Multifunctional Gestures. The gestures presented so far have been considered simple, and mainly focused on one control output. Combining different gestures will show the true potential of gesture-based interaction. Being able to control a complex control system with single gestures that control several outputs and then put in a sequence of non-overlapping gestures can be very valuable.

7. Rotating Cylinder with Flow. By combining the cylinder directional gesture with the gesture for flow control, one has a multifunctional gesture that gives the user the possibility to control flow and direction with one hand. Also in this case can the flow control be calibrated so that negative flow is possible (reverse etc.). The test of this gesture and visualization can be read in the next chapter.