

INPUT STRATEGIES FOR PRECISE CAD MODELLING WITH MODERN INTERACTION DEVICES

Markus Berger¹, Jörn Plönnigs¹, and Martin Wogan²

¹University of Rostock, Rostock, Germany

²TU Dresden, Dresden, Germany

Abstract

The release of ever more capable wearable display and interaction devices, like Apple's Vision Pro or the Meta Quest line of products, and the continuous change in the mobility of engineering work, enables new means of interaction with Computer Aided Design (CAD) software. This includes new input modalities that go beyond traditional mouse and keyboard interfaces. In this paper, we want to analyse these new input devices in the context of 3D CAD modelling tasks, and compare them to more conventional interfaces to see in which scenarios they may be most productive and where challenges lie on the way ahead.

Introduction

Computer Aided Design (CAD) was one of the initial innovation drivers behind the development of novel human-computer-interfaces like the mouse from Engelbart et al. (1965) or the stylus (light pen) by Sutherland (1964). Since then, the combination of mouse and keyboard established itself as the common way to model objects in CAD software. It allows designers and engineers to select objects, draw shapes and input numbers and text data quickly and with minimal movement affecting fatigue. However, Engelbart et al. (1965) never intended it to be used in 3D. Any modelling task that requires 3D inputs relies on either special key-combinations to add 3D translations and rotations, or on dedicated 3D-input devices, like 3D-mouses or trackballs, which add additional axes of inputs. Sans such devices, the 2D mouse is still the most common input device, simply because of its ease-of-use and due to the fact that most displays are still 2D.

This situation is shifting nowadays, with modern immersive display hardware, like VR (Virtual Reality), AR (Augmented Reality) or MR (Mixed Reality) headsets or high-quality stereoscopic 3D screens being more commonly available. Here, the models are actually shown in 3D space and mouse inputs quickly fail to compensate the missing third dimension. New specialized input devices become a necessity. Similar issues are observable with portable touch devices. Tablets and phones initially just emulated the mouse and keyboard and provided mouse-like touch inputs and virtual keyboards. However, most of the efficiency of these input methods vanished in practise. Specific multi-touch input gestures and swift keyboards developed, which proved more appropriate for most modelling tasks. But none of the finger-based touch interaction concepts have become established in CAD, primarily due to their inherent precision issues (Benko and Wigdor, 2010).

AR and VR systems on the other hand are recently experiencing a move away from physical controllers and towards fully hand-based interactions. These hand tracking systems are either used to simulate a full hand within the virtual space, in which the physicalized live shape of the hand is pressing buttons and touching elements (Meta Quest), or they recognize specific hand gestures and translate these into space, for example by combination with eye gaze (Apple Vision Pro, Microsoft HoloLens). While selection tasks and other simple interactions can be effectively represented, inputs that require high precision can be an issue. Additionally, the more specific interaction techniques are, the more difficult it is to achieve broad support for them in 3D CAD software.

Nevertheless, 3D headsets and touch devices have huge potential to enable new approaches to immersive design, where people in different physical locations can collaborate on a virtual model, and make changes by interacting with virtual objects through the same direct means they would use with physical models. How intuitive would it be if we could move weight bearing pillars in CAD as simply as furniture in the real world, in the process getting a feeling for the weight, the pressure, the sound absorption, and the touch of the material, and then still have the capability to position it down to millimeter precision?

To achieve this, we need to first solve the precision problem. Only once these novel interaction techniques can reliably achieve the same, or better, results than traditional interfaces will we see broad adoption in professional CAD software. For this, we have to learn from the dominant paradigms of mouse and keyboard and touch and stylus interfaces, which crucially are *bi-manual combinations* of two input modes that seems to net appropriate levels of precision. In the following sections, we want to answer the question of why that is, and how we can emulate this for arbitrary use cases involving any type of display and interaction device. To achieve this, we do the following:

- We discuss related literature from engineering design and Human-Computer Interaction (HCI)
- We define what common types of inputs and input devices exist and what their important attributes are
- We evaluate these types and devices according to several input metrics
- We apply these insights to three 3D CAD use cases
- We describe possible avenues of research that could lead to better CAD input systems

State of the Art

The question of which 3D input devices are useful for modelling tasks predates the advent of current generation AR and VR. For example, Fiorentino et al. (2010) states that “while planar features modelling performs well with mouse and keyboard, Direct Modelling benefits from bi-manual interaction, six degrees of freedom input (6DOF) and 3D snapping.” They also already distinguish between use cases, where a VR system for conceptual design has different input requirements than mechanical CAD modelling on a 2D screen. For the latter task, they consider the effectiveness of different 6DOF haptic input devices for 3D rotations, and which input methods are most effective for them. In this case, devices are distinguished by aspects like the shape of their transfer function from device to display or how the device responds to muscle force being applied to it.

The literature also features many other input-system specific measures of accuracy. There are attempts to improve typing speed on keyboards (Jiang et al., 2020), studies about the ideal way to implement pointing tasks with VR controllers (Allgaier et al., 2022), accuracy of gaze-tracked inputs (Schuetz and Fiehler, 2022), and much more. Most of these papers focus on individual tasks. Overviews of how these systems fare over all task in multi-input device for multiple purposes are much rarer.

More direct quality metrics used in HCI research are the concepts of resolution and throughput (Bérard et al., 2011). Resolution describes the smallest displacement that an input device can measure, while throughput describes the average rate of information generated by a series of movements (Fitts, 1954).

In addition to these often well-researched common input devices, various dedicated experimental interfaces exist in the literature. Lemberski and Hemmerling (2010) turned an audio mixing device into a dedicated CAD controller, in which 10 faders control 10 parameters in the CAD modelling process. The advantage of a physical interface like a fader, is that their physical position reflects their absolute value. Sharma et al. (2011) created a multimodal system in which touch-based menu controls are replaced by voice commands, and the modelling is done with touch inputs. Specifically, this is about using voice and touch in the early design stages, when exact values are not necessarily that important. Research like this highlights that we may have been trained to think that precision is important above all, as the interfaces we currently use do not distinguish between early stage and late stage modelling precision. If all a keyboard can offer is precise numerical inputs, then precise numerical inputs is how we will approach modelling. Sharma et al. (2011) write that “it is commonly observed that designers prefer pen and paper for early stages of design”, which is most likely the case for this exact reason.

Parameters of CAD Input

While the literature features many precise quantitative measurements of precision and throughput, we want to

propose broader, qualitative categories like: Can this device be used in the field? Are precise inputs even possible? Does using this device for prolonged periods get tiring? For this, we first need to establish a more systematic understanding of CAD inputs that stay relevant across many classes of interaction devices. This will aid us in establishing quality metrics that can be used as the basis for creating CAD interaction systems for arbitrary use cases and display setups.

Classification of Input Tasks

Fiorentino et al. (2010) “consider CAD modelling as a sequence of picking and navigation (rotation) actions”. More generally, they define four basic classes of CAD input tasks: *navigation and travel*, *selection*, *manipulation* and *system control*. We will base our analysis on this categorization, but will differentiate between spatial manipulations and numerical manipulations. *Spatial manipulations* involve the placement of points, drawing and alteration of lines and surfaces by direct spatial input. *Numeric manipulations* involve the input of numbers in relation to some part of the CAD model, for example by constraining the length of a wall to a specific precise length. Both of these interactions can have different levels or precision, depending on input devices. A mechanical slider like found on MIDI boards could potentially offer very precise inputs, compared to a completely virtual touch slider, as there is an inherent amount of smoothing and inference when dragging a finger over a touch surface.

Figure 1 shows the input tasks with their associated set of common operations. The input types are arranged in two classes—those that have a direct reference to the model being designed, and those that are used to put the software context into the proper camera position and program state to enable the next direct modelling steps.

Each task can be resolved in different way, depending on the input devices. Specifically, we will distinguish three classifications of input devices, which we will discuss in relation to selection, spatial manipulation and numeric manipulation tasks respectively.

Spatial Reference

Selection tasks happen at different resolutions, from object-level to the level of individual points on a geometry. The aspect that distinguishes them from each other more than anything however, is whether we select points

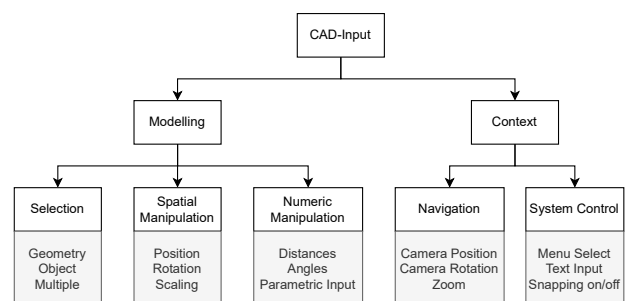


Figure 1: Hierarchical classification of CAD input tasks.

based on some projection to 2D (e.g. mouse with trackball), or whether the device can immediately reference 3D points independent of a display surface. Fig. 2 highlights this difference.

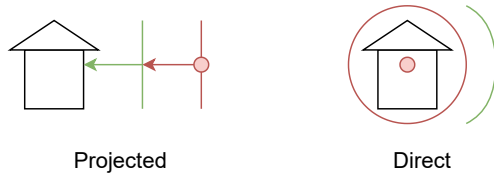


Figure 2: Green shows where the display is, red shows the interaction device, black denotes the object being selected.

Input Reference

Similar to selections, the way an input device does spatial manipulations will be most impacted by whether the device is creating a sensation of immediately interacting with the manipulated object, through a touch screen or in a mid-air 3D projection, or whether device movements happen on an unrelated surface or space and are somehow translated into the model. Figure 3 shows this distinction between translated and immediate inputs.

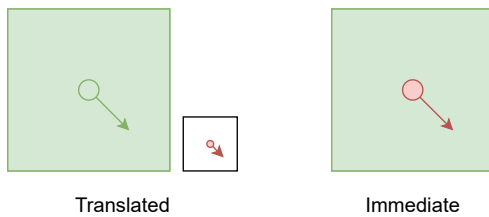


Figure 3: The green boxes represent a display device, the black box a separate zone of interaction. Red highlights the point where the input physically takes place.

Input Progression

Numeric inputs are distinguished by the basic fact of whether we are inputting the numerical value explicitly as a series of digits (usually to a high degree of accuracy), or whether we are providing a way to move as close as possible to it. The advantage of exact inputs is that the user can quickly set a-priori known values quickly—the disadvantage are case where the value is not known, like alignment cases. A system that allows for a continuous approach towards a value allows us to experiment with placements and sizes in a way that exact measurements do not. Though most relevant to numeric inputs, the capability to input numbers can have an impact on every kind of interaction, for example to select the scale at which a specific spatial operation should be resolved. Figure 4 highlights the process of these inputs over an imagined number line.

Input Devices

The main reference device for our analysis is the traditional combination of **mouse** and **keyboard**. The mouse usually acts as a combined selection (“Take *that*...”) and spatial manipulation input (“...and put it *there*...”) with reasonable accuracy, often supported by acceleration features. Meanwhile, the keyboard is used for numerical ma-

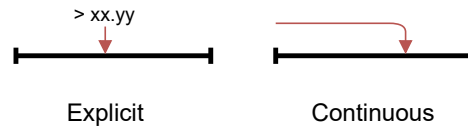


Figure 4: Numeric inputs are often collected by linear interfaces (e.g. slider). A device either moves from one side to the other (red line), or snaps to specific value on the line (red arrow)

nipulation (“...*there* being defined as *exactly*...”), but, often also provides alternative selection and manipulation key combinations. Individually and in tandem they also provide easy navigation and system control inputs. As stated in the introduction, this does not translate efficiently to 3D displays. This lack of good input devices is one of the main reasons, why we do not design in 3D¹. Regardless of any other considerations, any alternative input system must be able to offer at least a degree of precision and throughput that gets reasonably close to what a mouse and keyboard can achieve. What is “reasonable” depends on the specific use case, as will be discussed later on in the paper.

The class of touch devices is one potential alternative interface technology. Here we use a pointing device (e.g. a stylus or our fingers) often on a surface (e.g. screen, touchpad, tablets) with different gestures (e.g. tap, one-finger slide vs. two-finger drag) to realize various inputs. The most useful distinction for our purpose here is simple **touch** vs. **multi-touch** with the finger modality, and then specific **touch devices** like a stylus, which can also be combined, like shown in Pfeuffer et al. (2021). The benefit of those inputs is that they are very intuitive and easy to learn. Traditionally those interfaces were 2D, but, nowadays are sometimes translated into the 3D world by devices like the **3D-stylus** by Jackson (2020). Sometimes, a 2D stylus is also used on a specific, external **drawing tablet**. When used like this, it is similar to a mouse, but, has additional sensors for pressure.

Then, there are devices that use more novel input systems with a touch-metaphor. This is the case for both the Meta Quest (menu controls often use hand movements as a touch-analog, like the swipe of the hand modelling a finger drag) and the Vision Pro, which both offer gesture control modes. In case of the Vision Pro, a combination of **gaze** and gestures enacts the same types of inputs like Apple touch devices, thus enabling cross-platform compatibility. The main distinction here is between **direct gestures**, which are done in direct contact with an object, and **indirect gestures**, which are done somewhere in space and area in some way related back to the object in focus. Apple (2024) makes this distinction explicit in their developer guidelines. Indirect gestures are those that work in tandem with gaze to implement common touch gestures from other platforms: Tap, Swipe, Drag, Touch (or pinch) and

¹The lack of abstraction is another one. 2D plans actually enforce a simplification that is removing visual clutter and allowing users to focus on the relevant visual information. However, we know how to generate good, simple 2D views from 3D. So, it is technically a solved issue.

hold, Double tap, Zoom, Rotate. Direct gestures are a new addition specific to their AR platform and include the following: Touch, Touch and hold, Touch and drag, Double touch, Swipe, Pinch and drag two hands together or apart, Pinch and drag two hands in a circular motion.

A more specialized type of interaction hardware are controllers of any kind. These can include game controllers or even devices like MIDI controllers. These have in common that they combine multiple smaller input elements into a usually very ergonomic device. They feature devices like **buttons and triggers** (with just a few, sometimes analog buttons, as opposed to the many buttons on a keyboard), **analog sticks**, and **sliders and knobs**. These types of controllers are usually not tracked in space. Buttons and triggers are also usually part of other devices and are not deployed on their own. For the purposes of this paper, the class of analog sticks also encompasses any static 6DoF (Degrees of Freedom) controller, like 6DoF joysticks or 3D mouses.

A tracked extension of them are **motion controllers**, today mainly used for VR. These reduce the controller paradigm to what can comfortably be used in one hand, and add object and sometimes hand tracking to it. As such they model human hands, but other than direct hand gesture inputs they offer some translated inputs like analog sticks and triggers. Some motion controllers eschew additional controls and are instead focused on tracking and input accuracy—these types of controllers are equivalent to the aforementioned 3D stylus.

Some older VR controllers as well as controllers for room scale large display systems are only tracked in rotation to implement a laser pointer-like interaction. We will call this class of devices **3D pointing device**.

Finally, there is **Voice** control. Although this modality has fallen in and out of favor over the years, there could be great potential in using it in tandem with manual devices, specifically when these devices occupy both hands and can not by themselves cover every needed input.

Evaluation

In this section, we will take the previously established input types and devices, as well as their categorizations and evaluate them over some common metrics. This will yield a qualitative overview of which devices are appropriate for which use cases and how they can be combined.

Metrics of CAD Input

We first define some metrics to compare the different input devices in applicability to CAD design. Our metrics are based on the common quantitative metrics resolution and throughput defined for example in Bérard et al. (2011). *Resolution* is defined as the smallest possible displacement of an input device that can be attained reliably and quickly (i. e. with one short motion). *Throughput* describes, simply said, the average information an input device is capable of creating for the modelling task by a trained user. We distinguish in spatial throughput for spatial inputs, and

numerical throughput for numerical inputs.

We redefine these metrics as *Precision* and *Capability* to make them qualitatively evaluateable and extend them by *Explorability* as follows:

Precision (Resolution) How precisely can we input the information into the model and how error-prone are inputs?

Capability (Throughput) How quickly can we get the needed information into the model and how much range do we have for the inputs?

Explorability Can we quickly try out variations while we input information into the model, for example by moving a control back and forth?

All apply to both spatial and numerical inputs, though they refer to slightly different aspects for both. For Selections, Navigation and System Control, we only consider a generalized *Effectiveness* metric. Explorability is a secondary metric that derives from both the throughput of a device and which input classes it falls into.

Apart from the direct input metrics, there are also metrics set by the context of use. Do we need a portable device? How many hands do we have free? Are we building robust systems for experts or trying to invite the general public to play around with modelling? Secondary metrics are:

Hands Is an interaction uni- and/or bi-manual?

Intuitive Are the interaction concepts intuitive?

Portable Can the input device quickly be used in the field?

Collaboration Does the input device allow for dynamic collaboration of multiple in one room, on the same reference display or projection?

Comfort Is the device ergonomic, and is using it not tiring even when used for hours on end?

Support How well is this type of device supported in common 3D CAD software today?

Factors like portability, software support can usually be easily determined based on experience with how these products are used in practice. Intuitivity and comfort are more complex psycho-physiological measures, but are still commonly studied metrics in the HCI literature (e.g. in Stern et al. (2008) and others referenced in Table 1).

Comparison of CAD Input Devices

With these types of inputs and metrics of quality established, we can consider which input devices are effective for which kind of inputs. A qualitative classification of this can be found in Table 1. We first compare all input devices based on the use case metrics and specify whether the device intuitivity, portability, collaborativity, comfort, and current support are low (♥), medium (♦), or high (▲). We then specify in which spatial reference, input reference and numeric input the device falls and evaluate the effectiveness of the device to execute selection, spatial inputs and numeric inputs. For the latter two we evaluate the precision, capability, and explorability individually and for the first one the overall effectiveness. We do the same for the effectiveness for navigation and system control. These evaluations are, where possible, based on results from the lit-

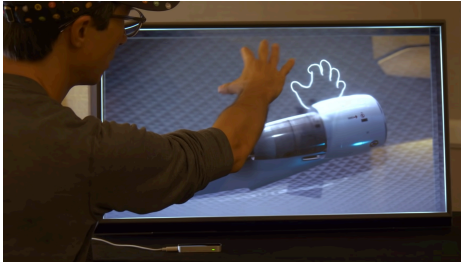


Figure 5: User interacting with a holographic display.
Source: Looking Glass

erature. Example references are given at the end of the table. This table can be used to prioritize the quality measurements important for a specific use case, and quickly compare and identify appropriate input devices. Input devices can also be combined in any way until both hands are full, or if they are portable enough.

There are of course other possibilities than the ones shown in the table, like direct numerical inputs with the mouse by clicking on a virtual keyboard. However, we want to consider only inputs that are actually common in practice or could possibly become important as new display and interaction modalities are unlocked through AR and VR technologies. Many input methods allow for the use of virtual keyboard, but they will usually have lower precision (because of being error prone) than a physical keyboard, as well as much lower capability, and as such have never gotten a large amount of adoption.

Discussion

In order to show how to use this evaluation in practice, we will consider three input scenarios that are not currently common in CAD modelling, but could become more commonplace in light of recent technological advances.

Scenario 1: Single person, 3D Display

This use case is aimed at a trained user drawing up an entirely new model from scratch on a CAD workstation with a holographic or high-quality stereo display, capable of displaying models with 3D depth, or even fully volumetric displays as shown in Fig. 5. The user can see all input devices and there is no occlusion, as will be discussed in the second scenario.

The most important input types here are spatial, numeric and system control, and within the former two both precision and capability play a large role. As far as the secondary metrics go, the most important one is comfort, as expert users in such a system would be expected to work for hours uninterrupted.

According to Table 1, the following devices would be appropriate choices to fulfill all or some of the requirements:

- Mouse/Drawing Tablet (For translated inputs and system control)
- Keyboard (For numeric input and system control)
- 2D Touch Stylus (For immediate inputs)
- Sliders/Knobs (For continuous numeric inputs)
- Trigger/Buttons (For system control)

All of these devices can offer comfortable working conditions, depending on display-setup. For a touch stylus to be comfortable for hours, the display should be slanted beneath the user, something that may not be ideal when utilizing a 3D display. The issue with simply using mouse and keyboard is the fact that the display projects a 3D image. The 2D-based translated projection of the mouse will still work relatively well for selection tasks, however the translated spatial inputs are similarly projected into 2D, which will not be sufficient for a true 3D context. Instead, we need input with a high *comfort* level, a *direct* spatial reference and an *immediate* input reference. If we consider all options in Table 1, there is no combination that covers all three of these aspects perfectly. A 3D stylus would be the best option, but will not be comfortable for hours, even with for example an elbow resting on the table in front of the screen. In lieu, we could combine a mouse or drawing tablet for precision spatial inputs on the model in the current camera view, and the ability for the user to reach into the model with direct gestures whenever necessary. This way we achieve the direct and immediate inputs, without having to regularly grab a new input device.

To promote the gesture paradigm, we could include indirect gestures for system control, thus making the weak hand the “gesture” hand while the right hand is kept on the mouse. This only creates an issue in that it would be awkward to have a full keyboard in the same space in which the hand is frequently moving down onto the table for indirect gestures and up from the table for direct gestures. Here, we could instead use a special control board with sliders, knobs or buttons with the left hand, to do additional system control and to input continuous numeric values whenever necessary. If a direct numeric input is necessary, the user can use voice control to say the exact numbers needed.

Scenario 2: Single person, VR Glasses

In this input scenario we imagine an architect or civil engineer working on a CAD model that features drastic scale differences in individual parts. The user needs to be able to change camera position and scale within the world on the fly, in order to place themselves in front of a part, while also seeing as much of the surrounding context as possible. Because of the limited virtual view that a 3D display would offer, we need to utilize a VR headset to properly display the model context. This creates an issue with the input devices we used in Scenario 1: they would now be fully occluded by the virtual world. This means that the user cannot see their interaction with the input devices and could potentially lose their positioning after looking away from their focus point. Thus, all input devices that we use in this scenario should be tracked in 3D space, in order to place avatars of them into the virtual space. This is most feasible with any input device that has an immediate input reference and direct spatial reference, as well as any type of input that does not require a device at all:

- Gaze (Selection)
- VR Motion Controllers (For spatial and numerical

Table 1: Comparison of various input devices in their effectiveness for certain CAD editing tasks. For spatial and numerical inputs, effectiveness is split into the three quality measures precision/capability/explorability. We evaluate in: \blacktriangleright - low, \blacklozenge - medium, \blacktriangleleft - high effectiveness.

Input Devices	Use Case Metrics					Modelling Inputs					Context Inputs		Example Ref.		
	Hands	Intuitive	Portable	Collab.	Comfort	Support	Spatial Ref.	Select.	Input Ref.	Spatial	Input Prog.	Numeric		Nav.	Sys. Ctrl.
2D Mouse	1	\blacklozenge	\blacktriangleright	\blacktriangleright	\blacktriangleleft	\blacktriangleleft	Projected	\blacktriangleleft	Translated	$\blacktriangleleft/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	\blacklozenge	\blacktriangleleft	Radhakrishnan et al. (2013), Besançon et al. (2017)
3D Mouse/Analog Stick	1	\blacktriangleright	\blacktriangleleft	\blacktriangleright	\blacktriangleleft	\blacktriangleleft	-	-	Translated	$\blacklozenge/\blacklozenge/\blacktriangleleft$	Continuous	$\blacktriangleright/\blacktriangleleft/\blacktriangleleft$	\blacktriangleleft	\blacktriangleright	van Berkel et al. (2022)
Keyboard/Buttons	1 - 2	\blacklozenge	\blacktriangleright	\blacktriangleright	\blacktriangleright	\blacktriangleleft	-	-	-	-	Explicit	$\blacktriangleleft/\blacktriangleleft/\blacklozenge$	\blacktriangleleft	\blacktriangleleft	Lepours (2018)
Simple Touch	1	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacklozenge	Projected	\blacklozenge	Immediate	$\blacktriangleright/\blacklozenge/\blacktriangleleft$	Continuous	$\blacktriangleright/\blacklozenge/\blacktriangleleft$	\blacklozenge	\blacklozenge	Besançon et al. (2017)
Multi-Touch	1 - 2	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacklozenge	Projected	\blacklozenge	Immediate	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacktriangleleft	\blacklozenge	Radhakrishnan et al. (2013), Tuddenham et al. (2010)
2D Touch Stylus	1	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacklozenge	\blacklozenge	Projected	\blacktriangleleft	Immediate	$\blacktriangleleft/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacklozenge	\blacklozenge	Romat et al. (2021)
3D Stylus	1	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacklozenge	\blacktriangleright	Direct	\blacktriangleleft	Immediate	$\blacktriangleleft/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacklozenge	\blacklozenge	Allgaier et al. (2022), Cannavò et al. (2020)
Drawing Tablet	1	\blacklozenge	\blacktriangleright	\blacktriangleright	\blacktriangleright	\blacklozenge	Projected	\blacktriangleleft	Translated	$\blacktriangleleft/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	\blacklozenge	\blacktriangleleft	Romat et al. (2021)
Gaze	0	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	Projected	\blacktriangleleft	Immediate	$\blacktriangleright/\blacklozenge/\blacklozenge$	Continuous	$\blacklozenge/\blacktriangleright/\blacklozenge$	\blacktriangleright	\blacktriangleright	Schertz and Fischer (2022), Bigman (2017)
Indirect Gestures	1 - 2	\blacklozenge	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	Projected	\blacklozenge	Translated	$\blacktriangleright/\blacktriangleright/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacktriangleleft	\blacktriangleleft	Lepours (2018)
Direct Gestures	1 - 2	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	Direct	\blacktriangleleft	Immediate	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacklozenge	\blacklozenge	Allgaier et al. (2022)
Sliders/Knobs	1	\blacktriangleleft	\blacklozenge	\blacktriangleright	\blacktriangleleft	\blacklozenge	-	-	-	-	Continuous	$\blacktriangleleft/\blacklozenge/\blacktriangleleft$	\blacktriangleright	\blacktriangleright	Tuddenham et al. (2010), van Berkel et al. (2022)
Trigger/Buttons	1	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	\blacktriangleleft	\blacklozenge	-	-	-	-	-	-	\blacktriangleright	\blacktriangleleft	Lepours (2018)
3D Pointing Device	1	\blacklozenge	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	Direct	\blacktriangleleft	Immediate	$\blacktriangleright/\blacklozenge/\blacktriangleleft$	Continuous	$\blacklozenge/\blacklozenge/\blacktriangleleft$	\blacklozenge	\blacklozenge	Mohr et al. (2019)
Voice	0	\blacktriangleleft	\blacktriangleleft	\blacklozenge	\blacklozenge	\blacktriangleright	-	-	-	-	Explicit	$\blacktriangleleft/\blacklozenge/\blacktriangleright$	\blacktriangleright	\blacktriangleleft	Lepours (2018)
Composite Devices															
VR Motion Controller	1	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	\blacklozenge	Direct	\blacktriangleleft	Immediate	$\blacklozenge/\blacklozenge/\blacktriangleleft$	Continuous	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	\blacktriangleleft	\blacklozenge	Allgaier et al. (2022), Cannavò et al. (2020)
Gaze and Gesture	1 - 2	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleleft	\blacktriangleright	Projected	\blacktriangleleft	Immediate	$\blacktriangleright/\blacktriangleleft/\blacktriangleleft$	Continuous	$\blacklozenge/\blacktriangleleft/\blacktriangleleft$	\blacktriangleleft	\blacklozenge	Lysbaek et al. (2022)

Table 2: A mapping of input tasks to predefined indirect gestures from Apples Vision Pro (Apple, 2024) and voice control.

Input	Gesture or device
Part Selection	Gaze and Tap
Navigation	Drag with one hand, Pinch and drag two hands in a circular motion
System Control	Gaze and Tap, Double Tap, Swipe
Moving and Altering Parts	Pinch and drag apart with two hands, Voice



Figure 6: VR headset and controllers in a CAD model. (Arch Virtual)



Figure 7: User interacting with a Apple Vision Pro. (Apple)

manipulation, navigation)

- 3D stylus (for precise spatial manipulation)
- Direct Gestures (Navigation)
- Indirect Gestures (For system control)
- Voice (For numeric inputs and system control)

The occlusion removes several interfaces that strongly depend on a good hand-eye-coordination like the mouse, keyboard or 2D stylus devices from our list. VR motion controllers or 3D styluses are better suited as they can be synchronized in the 3D model and users have a hand-eye-coordination feedback loop as shown in Fig. 6. One issue is that none of these devices offer high precision *and* a high comfort level. It is however unlikely that a user would wear VR glasses for a complex, multi-hour interaction process. Instead, this system is more likely to be deployed only once a high degree of spatial awareness is necessary. If longer working times are expected, we could in a first step either create more involved indirect gestures that also allow for numeric inputs, or for long-term work even create a custom tracked solution based on sliders, knobs and buttons.

Scenario 3: Collaborative, VR/AR Glasses

For this scenario, we imagine a group of designers doing design ideation around the same 3D model. They can twist and turn the model as well as highlight and switch out parts. The most important aspects here are selection, navigation and spatial manipulation. Hand-free interactions are more relevant than precision for this. While the input reference should be immediate, a direct spatial reference is not absolutely necessary. In this collaborative setting, users are expected to sit around a table in a relaxed manner, and will thus be relatively far from the model.

- Gaze (For selection)
- Indirect Gestures (For navigation and spatial manip.)
- Voice (for direct numeric input and system control)

Until recently, Microsoft’s HoloLens display was the most prominent system that could implement this combination of inputs. However, display quality and depth of interaction was limited. Recently, the Vision Pro glasses by Ap-

ple have further developed this paradigm. What the Vision Pro offers us is a “Gaze and Gesture” input system as shown in Fig.7. Following Table 1, it is fit for comfortable collaboration, and can deliver reasonably well on selection, spatial manipulation and navigation tasks. We suggest a mapping of indirect gestures to common tasks in Table 2. The only issue is the precision of gestures in spatial manipulation. Placing parts in the model that is several meters away by hand gestures can be jittery and error prone. As such, these inputs should be transformed into step-wise numerical inputs with indirect gestures that mirror multi-touch gestures, and optional voice input for the highest precision.

Summary and Future Work

In this paper we review various classes of currently available input devices for CAD modelling applications. We identified commonalities and differences between these devices and categorized them according to multiple factors. Then, we established qualitative metrics for evaluating their expected performance for different input tasks. This being early stage research, the considerations were not validated quantitatively in a usability study. This is future work. We then applied these metrics to three use case scenarios that could profit from making use of novel display and interaction devices, discussing individual drawbacks of the device classes in context, and referencing specific actually existing devices where appropriate.

Establishing new input modalities is as much a process of breaking habits as it is of building effective devices. The issue that most new devices encounter, is that mouse and keyboard do most things very well, and specific 3D tasks reasonably well. A new device that does better at specific 3D modelling tasks will always have to compete with the broad usefulness and software support of paradigms like mouse and keyboard. Specific devices will be limited to specific software, a circumstance that can create enough inertia that even useful techniques do not end up in main-

stream CAD software. However, in the near future we can expect a broader adoption of VR and MR devices—which has the capacity to break some of this technological inertia and thus introduce whole new device classes into the public mind. CAD software should prepare for this now, in order to exploit entirely novel possibilities offered by some of these technologies. There is potential for systems that make it easier to draw organic shapes for 3D printing applications, or inputs that natively include AI-assisted interactions, combining the precision of a manual stroke with the *intent* behind it.

References

- Allgaier, M., Chheang, V., Saalfeld, P., Apilla, V., Huber, T., Huettl, F., Neyazi, B., Sandalcioglu, I. E., Hansen, C., Preim, B., et al. (2022). A comparison of input devices for precise interaction tasks in VR-based surgical planning and training. *Computers in Biology and Medicine*, 145:105429.
- Apple (2024). Gesture specification - standard gestures. <https://developer.apple.com/design/human-interface-guidelines/gestures>. Accessed: 29.01.2024.
- Benko, H. and Wigdor, D. (2010). Imprecision, inaccuracy, and frustration: The tale of touch input. *Tabletops-Horizontal Interactive Displays*, pages 249–275.
- Bérard, F., Wang, G., and Cooperstock, J. R. (2011). On the limits of the human motor control precision: the search for a device’s human resolution. In *Int. Conf. Human-Computer Interaction*.
- Besaçon, L., Issartel, P., Ammi, M., and Isenberg, T. (2017). Mouse, tactile, and tangible input for 3D manipulation. In *CHI - Conf. on human factors in computing systems*, pages 4727–4740.
- Blignaut, P. (2017). Using smooth pursuit calibration for difficult-to-calibrate participants. *J. Eye Movement Res.*, 10(4).
- Cannavò, A., Calandra, D., Kehoe, A., and Lamberti, F. (2020). Evaluating consumer interaction interfaces for 3D sketching in virtual reality. In *Int. Conf. on Arts, IT, Interactivity and Game Creation*, pages 291–306.
- Engelbart, D., English, W., and Huddart, B. (1965). Final report: Computer-aided display control. Technical report, Stanford Research Institute.
- Fiorentino, M., Uva, A. E., Fabiano, M. D., and Monno, G. (2010). Improving bi-manual 3D input in CAD modelling by part rotation optimisation. *Computer-Aided Design*, 42(5):462–470.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381.
- Jackson, B. (2020). Ovr stylus: Designing pen-based 3D input devices for virtual reality. In *IEEE Conf. on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 13–18.
- Jiang, X., Li, Y., Jokinen, J. P., Hirvola, V. B., Oulasvirta, A., and Ren, X. (2020). How we type: Eye and finger movement strategies in mobile typing. In *CHI Conf. on human factors in computing systems*, pages 1–14.
- Lemmerski, D. and Hemmerling, M. (2010). Mixer modeling—an intuitive design tool. In *Conf. on Education in Computer Aided Architectural Design in Europe*.
- Lepouras, G. (2018). Comparing methods for numerical input in immersive virtual environments. *Virtual Reality*, 22(1):63–77.
- Lystbæk, M. N., Rosenberg, P., Pfeuffer, K., Grønbæk, J. E., and Gellersen, H. (2022). Gaze-hand alignment: Combining eye gaze and mid-air pointing for interacting with menus in augmented reality. *ACM on Human-Computer Interaction*, 6(ETRA):1–18.
- Mohr, P., Tatzgern, M., Langlotz, T., Lang, A., Schmalstieg, D., and Kalkofen, D. (2019). Trackcap: Enabling smartphones for 3D interaction on mobile head-mounted displays. In *CHI - Conf. on Human Factors in Computing Systems*, pages 1–11.
- Pfeuffer, K., Dinc, A., Obernolte, J., Rivu, R., Abdrabou, Y., Shelter, F., Abdelrahman, Y., and Alt, F. (2021). Bi-3d: Bi-manual pen-and-touch interaction for 3d manipulation on tablets. In *An. ACM Symp. on User Interface Software and Technology*, pages 149–161.
- Radhakrishnan, S., Lin, Y., Zeid, I., and Kamarthi, S. (2013). Finger-based multitouch interface for performing 3D CAD operations. *Int. J. of Human-Computer Studies*, 71(3):261–275.
- Romat, H., Fender, A., Meier, M., and Holz, C. (2021). Flashpen: A high-fidelity and high-precision multi-surface pen for virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 306–315.
- Schuetz, I. and Fiehler, K. (2022). Eye tracking in virtual reality: Vive pro eye spatial accuracy, precision, and calibration reliability. *J. Eye Movement Res.*, 15(3).
- Sharma, A., Madhvanath, S., Shekhawat, A., and Billingham, M. (2011). MozArt: a multimodal interface for conceptual 3D modeling. In *Int. Conf. on multimodal interfaces*, pages 307–310.
- Stern, H. I., Wachs, J. P., and Edan, Y. (2008). Designing hand gesture vocabularies for natural interaction by combining psycho-physiological and recognition factors. *Int. J. of Semantic Computing*, 2(01):137–160.
- Sutherland, I. E. (1964). Sketch pad a man-machine graphical communication system. In *SHARE design automation workshop*, pages 6–329.
- Tuddenham, P., Kirk, D., and Izadi, S. (2010). Graspables revisited: multi-touch vs. tangible input for tabletop displays in acquisition and manipulation tasks. In *SIGCHI - Conf. on human factors in computing systems*.
- van Berkel, N., Merritt, T., Bruun, A., and Skov, M. B. (2022). Tangible self-report devices: accuracy and resolution of participant input. In *Int. Conf. on Tangible, Embedded, and Embodied Interaction*, pages 1–14.