

8

Interaction Devices and Techniques for 3D Sketching

*Mayra Donaji Barrera Machuca, Rahul Arora, Philipp Wacker,
Daniel Keefe, and Johann Habakuk Israel*

In the previous chapter, we discussed the different ways to create a stroke. In this chapter, we present various interaction devices and techniques that help users sketch in a 3D environment. An interaction device or technique is a combination of hardware and software elements that help users accomplish a single task [1], in this case, to create 3D sketches. When talking about specific interaction devices or techniques, it is important to consider the needs of the user. For 3D sketching, these needs change depending on the type of sketch the user wants to create [2, 3, 4]. It is also important to consider the affordances that different interaction devices or techniques have, as they affect the way users utilise them [5, 6]. Affordances are an attribute of the tool that shows which features they offer to the user [5, 6].

Based on these considerations, we divide this chapter into two parts. First, we will talk about user interface or interaction technique that allow users to draw conceptual and technical sketches. We will discuss the characteristics that makes them more appropriate for each specific sketch type. Then, we will talk about the functional and physical affordances of a user interface or interaction technique. Here, we will discuss how these affordances can affect the sketch produced by them. This chapter is useful for readers that want to design their interaction device or technique for sketching in 3D, as we discuss the different properties they need to consider. For example, the type of sketch their interaction device or technique can create and the affordances they need to have.

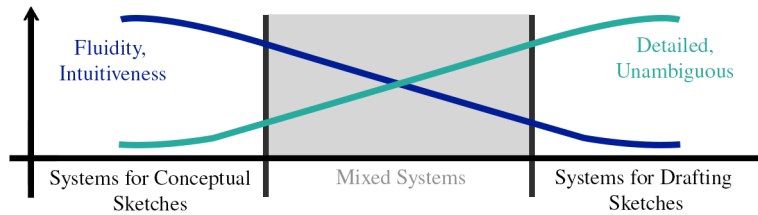


Figure 8.1: The three different types of systems we are discussing in this chapter.

8.1 Sketch types

In [Chapter 1](#), we already talked about the design process, which is “the complex activity of creating and evaluating specifications of artifacts whose form and function achieve stated objectives and satisfy specified constraints” [7] and the importance of sketching during it. In general, during the design process, the goal of the user for the sketches they make changes depending on their current stage, and this goal informs the sketch type [4]. For example, a designer produces a conceptual sketch when exploring a new idea, and a technical drawing when they are creating the final design of a product. Here, we divide the interaction devices and techniques into three groups ([Figure 8.1](#)) based on their functionalities:

- (a) systems for conceptual sketches, which are unstructured and ambiguous,
- (b) systems for drafting that produce accurate scale drawings,
- (c) mixed systems that help users draw sketches with characteristics of the other two sketch types

The systems in this last category are those that see 3D sketching as a unique medium and try to create a balance between keeping the interaction fluid and making accurate sketches.

8.1.1 Conceptual 3D sketches

In [Chapter 1](#), we state that conceptual sketches are used early in the design process to communicate the general aesthetic and to explore the technical aspects of a design. In this stage, designers need to explore initial ideas

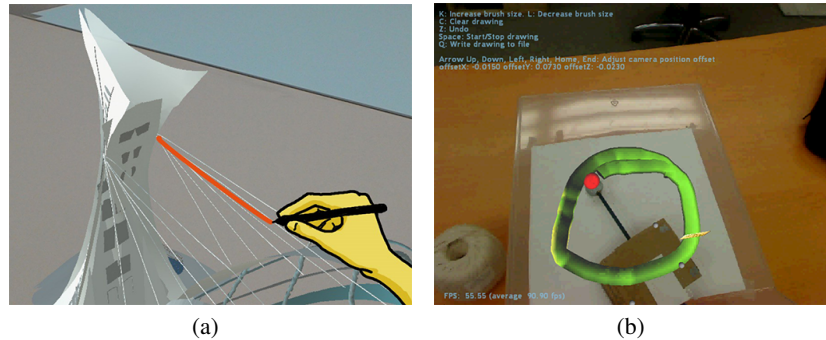


Figure 8.2: 3D freehand drawing systems follow the user hand movements. For example, (a) SymbiosisSketch [11] and (b) [12] system.

quickly and simply, so they can formulate a tentative solution. Designers of 3D sketching interfaces for conceptual sketches need to provide interaction techniques that are fluid, expressive and fast [8]. Interaction techniques in this category also need to be intuitive [9] and support the creative process, because other *windows, icons, menus, pointer (WIMP)*-based forms of interaction would fragment and hinder creative designing [10]. One interaction technique that meets these characteristics is freehand drawing, in which the stroke follows the user hand movement. There are different variations of freehand drawing, which depend on the input device used to create the strokes. Each of these implementations provides users with unique ways to draw more accurately.

3D freehand drawing

The first interaction technique we present is 3D freehand drawing which is the direct translation of sketching with a pen on a paper to 3D as illustrated in Figure 8.2. 3D freehand drawing uses a 6-DoF input device to follow the user's arm movements in space, which provides an intuitive and effective method of conceptualising new shapes [13]. This feature makes this interaction a popular choice among user interfaces [11, 14, 15, 16, 17, 18, 19, 20, 21, 12, 22, 23]. One characteristic of 3D freehand drawings is that the system does not constrain the user's actions. However, a user interface or device can use the stroke's properties to affect the sketch. To name three examples, in Fluid Sketching [17] users paint with fluids that

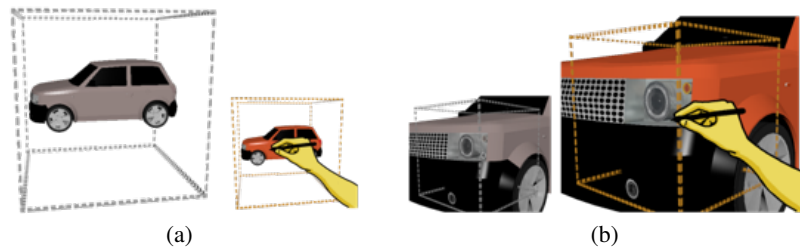


Figure 8.3: SymbiosisSketch [11] allows decreasing (a) or increasing (b) the object's scale to allow users to draw large-scale objects and fine details, respectively.

blend, in CavePainting different brushes have different behaviours [19] while Griddraw [20] creates a 3D grid of force vectors to alter the appearance of the stroke.

A limitation of 3D freehand drawing is the difficulty to control the stroke using the shoulder or the elbow [24, 25]. One way to solve this problem is to allow users to create small strokes that users can draw by only moving their wrist, and then provide tools to scale these strokes to their real size. Such an approach is adopted in SymbiosisSketch [11] and Tano et al.'s system as illustrated in Figure 8.3. However, these scaling techniques should not break the interaction flow by introducing WIMP-like interaction patterns to, for example, manually position the drawing plane. Another way to provide more stroke control, is to allow users to draw on a tracked 2D plane inside a 3D virtual environment [11, 15, 26]. These systems benefit from having a physical surface to draw on, and latch onto the user's experience of sketching with a pen. Similar to these interaction devices, other systems use a 2D screen to display the sketch [27, 28, 29]. However, these systems need to provide navigation tools so users can change their viewpoint. These systems also need to project the user 2D stroke to 3D. For example, Agile 3D Sketching [27] and SketchingWithHands [28] project the strokes to a virtual plane that is defined by the user. Both systems allow users to move their viewpoint using a combination of gestures.

3D modelling

There is also a category of user interfaces that use interaction devices and techniques to mimic the actions of modelling an object with clay [30, 31,

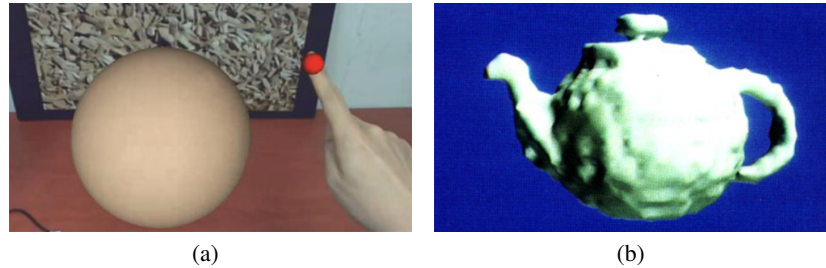


Figure 8.4: 3D modelling systems emulate the action of modelling with clay. For example, (a) AiRSculpt [33] and (b) Sculpting [31].

[32, 33, 34]. See Figure 8.4. These tools are more related to modelling than drawing, but they still focus on being fluid and expressive by not focusing on the creation of polygons. For example, AiRSculpt [33] allows users to model a sphere into different shapes by adding and removing material. Another example is Leal et al.'s system that provides users with tracked fabric which users can twist and reshape to create virtual surfaces.

In general, previously proposed interaction devices and techniques for 3D sketching conceptual sketches try to stay flexible and fast by constraining the user action the least amount possible. Designers of interaction devices or techniques for 3D sketching should think in ways to use the advantages of using a digital system, like Fluid Sketching [17] and Griddraw [20]. Also, finding novel ways to merge physical objects and virtual objects [32, 35] is still an open area of research.

8.1.2 Technical 3D sketches

Drafting sketches, also known as technical sketches, are mostly used during the later stages of the design process to show the final design of a product. In this stage designers need to precisely communicate an idea, so the sketch should be unambiguous and easy to understand. To achieve this, most designers use conventions and standards to represent objects, like the rules for orthographic projection. They also use CAD systems like AutoCAD [36] or Solidworks [37], as these allow to automate the process and bring a high degree of accuracy to the sketch but with the loss of fluid interaction. However, the control of the user interface requires much more attention from the designers than in sketch-based systems. Designers of 3D sketching

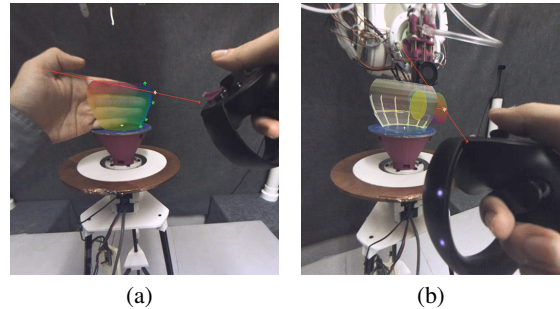


Figure 8.5: RoMA [38] is a two-modes system because it allows users (a) to use 3D freehand sketching and (b) to edit the vertices of a shape.

interactions for drafting sketches need to provide tools that allow users to create precise shapes like lines, circles and curves. User interfaces in this category also need to provide tools to edit the sketch. However, it is important to avoid creating a **WIMP**-based 3D modelling software and to avoid losing the intuitiveness and fluidity of sketching in 3D by constantly clicking on buttons or menu items.

Bimodal interfaces

One way to fulfil the requirements of drafting sketches is to create a user interface that has two modes, one uses 3D freehand drawing and the other allows users to create geometrical figures, like cubes, cylinders and spheres [39, 38] or use tools like revolve [40]. These two modes should be separated because the characteristics of each interaction are different. For freehand sketching, the ability to guide the interaction device should be intuitive, which does not require a conscious engagement. However, for creating specific geometric figures, the user needs to consider what each figure represents and its properties [41]. One example of such interfaces is RoMA [38], illustrated in Figure 8.5 that utilises 3D freehand drawing as the main interaction technique. The system also allows users to create traditional **CAD** primitives using revolve, extrude, loft, and sweep operations. By mixing both interaction techniques, the final product is accurate enough that a robotic 3D printer can fabricate the objects drawn by the user. Another example is Holosketch [39] which allows users to create strokes using freehand 3D drawing. Users can also create primitives like cylinders, spheres, and cubes.

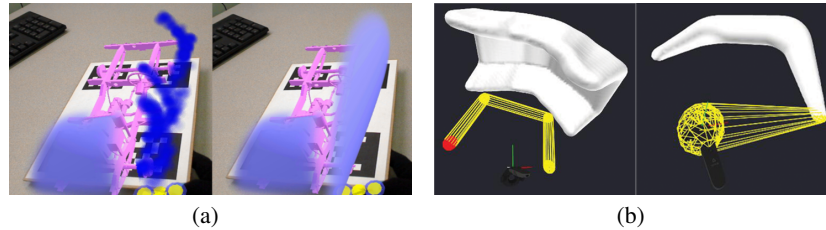


Figure 8.6: Automatic creation of 3D meshes systems use algorithms to translate the user hand movement in space into 3D meshes. For example, (a) Fuge et al.'s system and (b) Brush2Model [45].

By mixing both interaction techniques, users can create complex shapes without having to sketch every single object. In summary, these interfaces have one mode where users can sketch without constraints and another mode where users can reflect on what was drawn and refine it using widget-based interactions like handles and sliders.

Automatic creation of 3D meshes

Another way to create drafting sketches is to use an interaction technique that translates user movement in space into 3D meshes. In these interaction techniques, users create strokes using 3D freehand drawing. Then, the system translates these strokes into meshes [40, 42, 43]. Most systems aim to create the new surface without delay, but usually, the time between the stroke and the mesh creation depends on the computational power needed to calculate the new mesh and on the properties of the hardware used to run the system. One problem with this approach is that a noticeable delay between the two might affect the interaction by losing the fluidity of 3D freehand sketching. Other interaction techniques allow users to define the position and shape of a surface in space. For example, Fuge et al.'s system illustrated in Figure 8.6(a) automatically converts a cloud of points in space into surfaces. Users create these clouds by tapping with their fingers in the air using a custom-made glove to detect the hand gestures and the pressure of each tapping. Brush2Model [45], shown in Figure 8.6(b) gives users the ability to draw the skeleton of a 3D object. The system automatically covers this skeleton with a surface.

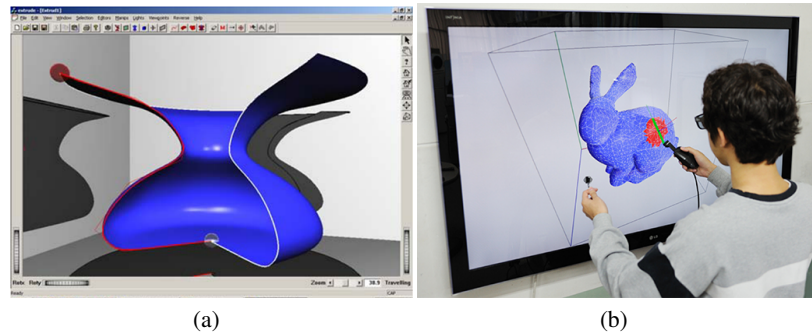


Figure 8.7: Systems for drafting sketches should allow users to edit their strokes. For example, (a) by directly editing the curve [46], and (b) by over-sketching the existing mesh [42].

Editing sketches

User interfaces for drafting sketches also allow users to edit their sketches. A common approach is to provide the standard tools used in modelling software, like tools to manipulate the vertices of the object [44, 42, 45], change the scale, position and orientation of the object [39], or manually modifying the curves [46] as illustrated in Figure 8.7(a). Another approach is to allow users to over-sketch the existing meshes [46, 42, 43], with this technique the user re-draws a new line over the previous one to correct any mistakes as illustrated in Figure 8.7(b). Giving users the ability to over-sketch their designs, allows them to fix their mistakes without breaking the sketch-like interaction, as over-sketching is widely used as a design tool for pen and paper drawing [47].

Designers of interaction devices or techniques for 3D sketching should consider different ways to use the advantages of sketching in 3D while simultaneously giving users tools to edit their strokes in-depth. This way, users can fix any error they made, like in RoMA [38] and SPACESKETCH [42]. Future interaction devices or techniques should also try to create systems that allow users to focus on the design process and less on how to create the surface [44, 45].

8.1.3 Mixed interfaces

Mixed interfaces aim to enable users to accurately sketch objects without losing the advantages of sketching in a 3D space, such as fluidity, expressiveness, and speed. These interfaces limit the user actions to make them more precise but do not follow the conventions and standards of desktop-based CAD. Mixed interfaces also rely on automation to remove the cognitive load of the user. The resulting sketch can be an intermediate step between creating a concept with pen and paper and a 3D model with a CAD tool. Designers of 3D sketching interactions of mixed interfaces need to focus on creating a balance between the need to be precise and how much they constrain users, as they might run the risk of creating an interface that limits the user's creativity [48, 49].

Strokes to surfaces

There are different interactions that achieve a balance between precision and fluidity. One of such interactions is strokes to surfaces [50, 51, 13]. These user interfaces allow users to create 3D strokes with different methods like freehand 3D drawing [13] or tape drawing [50]. In the tape drawing technique, users draw sketches on large scale surfaces using tape. Users unroll the tape with one hand and slide the other hand along with the tape while fastening it on the surface. Then, users extrude these strokes to create surfaces. For the 3D variation, users then extrude these strokes to create surfaces. See Figure 8.8(a) for a diagram of the interaction. An advantage of the strokes to surfaces interaction is that users control both the stroke and the surface creation process. However, users might make errors when sketching the strokes that they will only notice once they extrude them. To solve this problem, Lift-Off [51], illustrated in Figure 8.8(b) automatically creates strokes using a 2D sketch. Users then position these strokes in space before extruding them.

3D strokes to shapes

Other interfaces use the 3D strokes to shape interaction. In these interfaces, users draw strokes in space, and then the system automatically transforms these strokes into 3D models [52, 44, 53, 26] as shown in Figure 8.9. As with the strokes to surfaces interaction, different systems use various ways to create strokes. For example, 3D freehand drawing [52, 53] and hand gestures [44]. The difficulty of using the 3D strokes to 3D shape interaction is

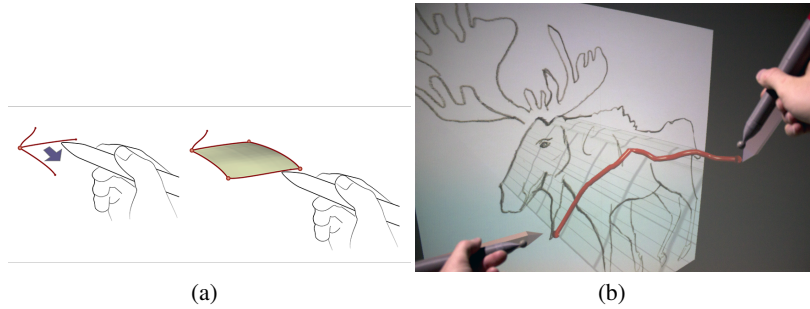


Figure 8.8: The strokes to surfaces interaction allow users to manually convert strokes into surfaces. (a) diagram of the interaction [51], and (b) interaction on Lift-Off [51].

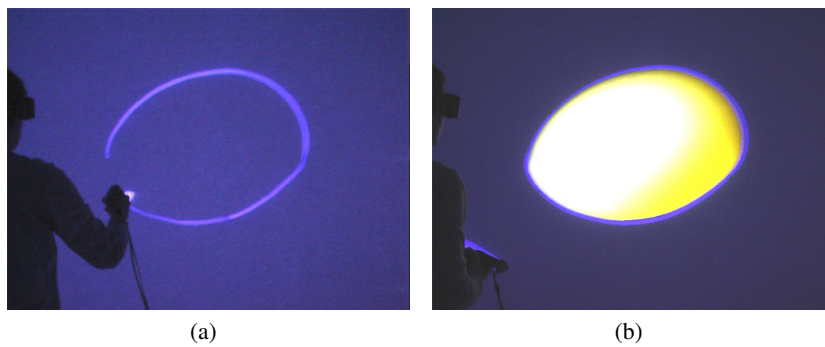


Figure 8.9: The 3D strokes to shapes interaction automatically convert the user strokes into 3D shapes. For example, in ImmersiveFiberMesh [53] (a) the user draws a stroke, and (b) the system automatically creates a 3D mesh.

to develop interpretative algorithms that are not overly sensitive to 3D input inaccuracies. One way to improve user accuracy is to make users draw the strokes in a physical object. For example, a tracked object that allows users to directly create 3D strokes [26].

2D strokes to shapes

A similar interaction technique that also uses physical objects is called 2D strokes to shapes [54, 55, 56, 51, 57, 58]. In this interaction technique,

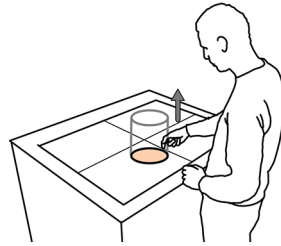


Figure 8.10: The 2D strokes to shapes interaction allow users to manually extrude 2D strokes to create shapes. A system that uses this interaction is (a) Mockup Builder [54].

users draw 2D objects that they later can extrude to 3D shapes in a similar way as to how some 3D CAD tools work [59]. For example, Mockup Builder [54, 55] shown in Figure 8.10 uses a touch table where users draw strokes. Then users use hand gestures to extrude the new surface. The user's action mimics pulling and pushing the shape out of the table, so it is easy to understand. This interaction technique also allows users to see the result of their extrude in real-time. User interfaces that use the “2D strokes to shapes” interaction techniques usually include tools to edit the sketch. For example, some user interfaces allow users to change the position of vertices, edges and surfaces [57]. Finally, other user interfaces edit sketches by over-sketching the 3D model [58].

Beautification

The last interaction technique in this category is beautification, which translates the user's informal strokes into structured shapes. This interaction technique automatically predicts the user's intended drawn object to transform the stroke into the correct shape. Beautification is widely explored in 2D systems [60, 61, 62, 63], but less in systems for 3D sketching [64, 65, 66]. For example, Multiplanes [64] (Figure 8.11) beautifies the user's stroke to lines and circles in real-time. Doing the beautification in real-time allows users to visualise the detected shape before they finish drawing, which allows users to change their hand movements accordingly. On the technical aspect, the main problem with beautification is to develop an algorithm that correctly identifies the user's intention. Otherwise, the user can get frustrated. However, new research on improved object recognition with neural networks [67, 68] is trying to solve this problem. On the interaction aspect,

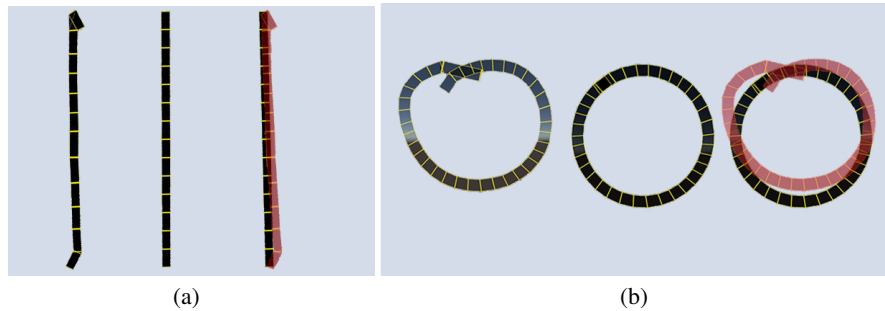


Figure 8.11: Beautification examples from Multiplanes [64]. (a) Lines, and (b) circles.

the problem with beautification is that beautified sketches can lose some of their authenticity and stroke expressiveness, for example, lines not meeting at corners or lines which are not straight [69]. Another problem with excessively precise sketches, is that it violates the principle of a draft, that “preliminary ideas should look preliminary” [70, 71].

Designers of mixed interfaces for 3D sketching should consider the level of assistance the user interface provides. For example, one decision designers need to make is if the user interface translates the user’s strokes to a clean, well-connected curve network, like FreeDrawer [13] or not, like Multiplanes [64]. Another decision is the amount of control the user has. For example, if the user explicitly controls the extrusion paths like in Lift-Off [51] or chooses from a fixed set of templates like in Window-Shaping [56]. All these decisions affect the type of sketch the user can create.

Finally, regardless of the sketch type the user interface creates, some design decisions depend on the type of device and interaction techniques used to draw the sketch, as different devices and interaction techniques provide different affordances. In the next section, we will discuss this topic in-depth.

8.2 Interaction techniques and devices affordances

Affordances are an attribute of the tool that shows which features they offer to the user [5, 6]. Therefore, depending on the interaction technique and the device used to create a sketch, their affordances affect the final sketch. For example, it is important to consider the tip width of a pen, as this will affect the level of detail of the sketch. Another feature to consider is the pen shape

and size. Designers of user interfaces for 3D sketching need to consider the affordances of the interaction technique and the device they choose. In this section, we focus on the functional and physical affordances described by [5].

8.2.1 Functional affordances

Functional affordances are a design feature of an interaction technique or device that helps users accomplish their work and are related to the way the tool works [5, 72]. For example, the grid in a graph paper provides designers with a functional affordance, as the presence and size of the grid affect the way the user draws. Interaction devices and techniques for 3D sketching have two main functional affordances. The first is the way users control the stroke creation process. The second are the additional tools the system provides to the user to help them be more accurate. Next, we are going to describe these two functional affordances of a user interface.

8.2.1.1 3D Stroke Creation

In the last section, we described 3D freehand drawing as an interaction technique where the user creates a stroke by moving their hand in space. However, this interaction technique presents control and precision issues related to the challenges of 3D sketching, see [Section 6.2](#) for more information. Here, we discuss different interaction techniques that affect the way the stroke follows the drawing tool position. These interaction techniques help users draw more accurately, but they also affect the look of the sketch. For example, the stroke length, that is, short versus long, or the line straightness, that is, curvy versus straight strokes. Finally, any of these stroke creation methods can be used in the sketch types discussed in [Section 8.1](#). However, it is important to note that user interfaces created for different sketch types can use any of the stroke-creation interaction techniques we are going to discuss next.

Follow-hand

The most common stroke creation interaction technique is follow-hand. This interaction technique is easy to understand and implement because it directly translates the action of drawing with a pen to 3D sketching as shown in [Figure 8.12](#). It also allows users absolute control over the resulting geometry, as it will follow all their hand movements. Examples of systems using the follow-hand interaction technique are Smart3DGuides [14],



Figure 8.12: Follow-hand is the most common stroke creation interaction technique. (a) CavePainting [19], and (b) diagram of the interaction [16].

SymbiosisSketch [11], and Multiplanes [64] among others [15, 17, 18, 23, 19]. Current commercial systems, like Tiltbrush [73], also use this interaction technique. It is important to consider the amount of control a user has over the stroke, as the sketch type will dictate the requirements. For example, for conceptual sketches, a stroke with uneven twists along the stroke might be desired, but for a final draft, some filtering needs to be applied to the geometry to prevent them. An example of a system that filters the strokes is SymbiosisSketch [11], which filters the user hand movement based on stroke dynamics (drawing velocity). See Section 7.1 for more information. Finally, it is important to consider that too much filtering can limit the characteristic appearance of a sketch as a sketch.

For the follow-hand interaction technique, the type of geometry used to represent the stroke affects the level of control the user has. In user interfaces that use calligraphic stroke, for example ribbons, the arm and the controller orientation affect the stroke, making it difficult to create smooth lines [74]. This problem is less present when the system uses cylinders because then, the controller orientation does not influence the stroke. See Chapter 7 for a technical explanation of surface creation. However, regardless of the geometry used, the follow-hand interaction is sensitive to the user's mistakes. Yet, as we mentioned before, this interaction technique also provides users with total control of the stroke in the way of changing the stroke orientation while drawing, which could make it preferred by experts.

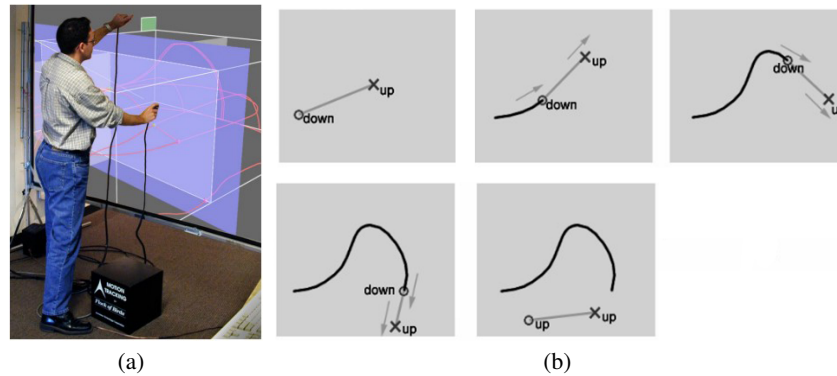


Figure 8.13: 3D tape drawing uses two hands to give users more control of the drawing stroke. (a) [50]’s system, and (b) diagram of the interaction [50].

3D tape drawing

Another stroke-creation interaction technique based on a 2D drawing technique is 3D tape drawing [18, 50, 74]. 3D tape drawing is a bimanual drawing technique that provides explicit control of the tangent of the drawn curve shown in Figure 8.13. In this interaction technique, the non-dominant hand defines the tangent, and the dominant hand draws the line following the tangent. Some implementations of 3D tape drawing use a drawing plane to project the stroke, as they directly translate the technique from 2D. For example, virtual planes on Grossman et al.’s system and physical planes on Fleisch et al.’s system. Other implementations, like [74], translate this method to 3D freehand drawing and do not project the stroke to a plane.

Compared to the follow-hand interaction technique, 3D tape drawing helps reduce noise in the arm movements. This interaction technique also helps users visualise the curve direction, which translates into better planning of the stroke. However, one problem of 3D tape drawing is that it requires coordination between the two hands. Another problem is that some shapes, such as circles, are difficult to draw. Finally, 3D tape drawing is commonly used when designing objects with curved lines, like cars.

Line-drag

The Line-drag interaction technique [50, 74, 75, 20] can be considered a one hand implementation of the 3D tape drawing interaction technique.

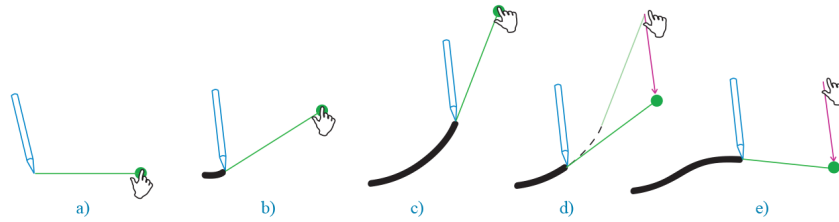


Figure 8.14: Line-drag is a one-hand implementation of 3D tape drawing. Diagram of the interaction [74].

When using Line-Drag, the user defines a path that the stroke follows, see Figure 8.14. A metaphor that better describes this interaction technique is dragging a rope with the stroke attached at the end. One characteristic of the line-drag interaction technique is that different implementations apply different rules to the way the stroke follows the line. The simplest implementation is to let users define the rope length. In Grossman et al.'s system, users first define the start point with a click. Then, users draw the path. Finally, with a second click, the stroke follows the drawn path. Another implementation is to automatically adjust the length of the space between the stroke and the cursor. For example, Drawing on Air [74] uses the stroke length to adjust this length. The system has a predefined minimum and maximum length to avoid errors. On the other hand, Dynamic Dragging [75] adjusts this length based on the expected curvature of the stroke, the drawing speed or a mix of both. A different approach is to use external constraints to affect the length between the stroke and the cursor. For example, Dynasculpt [20] uses a physical simulation that gives the cursor a specific mass, and it considers the path as a damped spring. In this implementation, playing with the cursor and path parameters affect the stroke properties.

The advantages and use cases of the line-drag interaction technique are similar to those of 3D tape drawing. However, line-drag solves the problems of using a bimanual interaction, like the coordination required to draw difficult shapes. The disadvantage of this interaction technique is to choose how the stroke follows the cursor: manual methods can be tedious, but automatic methods can confuse users.

Connecting points

Until now, we discussed stroke creation interaction techniques that emulate

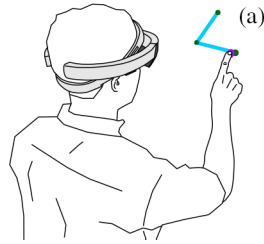


Figure 8.15: The connecting points interaction creates a line between vertices defined by the user. An example of this interaction is Tapline [16].

drawing with a pen and paper. Other interaction techniques automate the stroke creation process. For example, the connecting points interaction technique automatically creates a line between vertices defined by the user as shown in Figure 8.15. This interaction technique is aimed at drawing straight lines because users only need to define the start and end of the stroke. For drawing curves a system can use a spline curve, where the stroke interpolates between the points the user gives like Gravity Sketch [76] does.

There are different ways to place the vertices in space. One way is to directly position the vertex in space like Tapline [16] and Brush2Model [45]. Another way is to use an indirect way to position the vertices, like GoGo-Tapline [16], which uses the GoGo-cursor [77] to position the points outside the user's reach. Finally, connecting points can create both strokes [16] and surfaces [44].

Bimanual creation

Another interaction technique that also relies on automation is bimanual creation that allows users to draw using their two hands [51, 22, 45]. Bimanual creation uses two controllers, each attached to one end of the stroke. When using this interaction technique, users define the stroke curvature and length. Then the system translates these values into a stroke. The separation between controllers defines the stroke length, and to change the stroke orientation, users rotate the controllers. Bimanual creation allows users to create smooth curved strokes with many arcs or twists, which can be difficult with other stroke creation interaction techniques as illustrated in Figure 8.16.

Different levels of automation are possible. For example, Lift-Off [51] automatically creates the curves from 2D strokes, and the user only positions

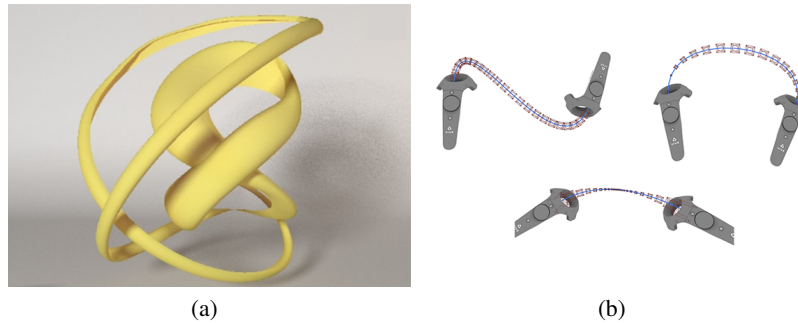


Figure 8.16: The bimanual creation interaction uses two controllers to define the curvature and length of a stroke. (a) [22]’s system, and (b) example of different strokes created by the different combinations of controller positions [22].

and rotates the curve. On the other hand, the system proposed by McGraw et al. [22] uses Hermite spline curves, and the user wrist orientation controls the curves of the stroke.

Physical-tools

The physical-tools interaction techniques [78, 31, 19] use the physical device shape to communicate the possible actions to the user and try to emulate the real-world capabilities of those devices virtually. For example, emulating the way paint reacts with gravity when thrown from a bucket or the stroke properties of a stroke made by a specific shape of a brush as illustrated in Figure 8.17. It is important to note that the name of these interaction techniques is a metaphor to describe the context the physical device communicates to the user and does not relate to the ergonomic properties of the device. The physical-tools interaction techniques’ main advantage is to bring familiar behaviours to 3D sketching. Physical-tools also allows users to explore other design possibilities besides lines and geometrical figures. For example, three brushes of Cave painting [19] enter this category. The Jackson Pollock++ and splat sprinkle a wall of the CAVE with paint, and the Bucket covers a whole surface with paint as if it was thrown with a bucket. Physical-tools also emulate specific tools found in the real-world. Three modelling tools of “Sculpting” [31] enter this category. The Additive Tool leaves a trail of material, like squeezing a toothpaste tube.

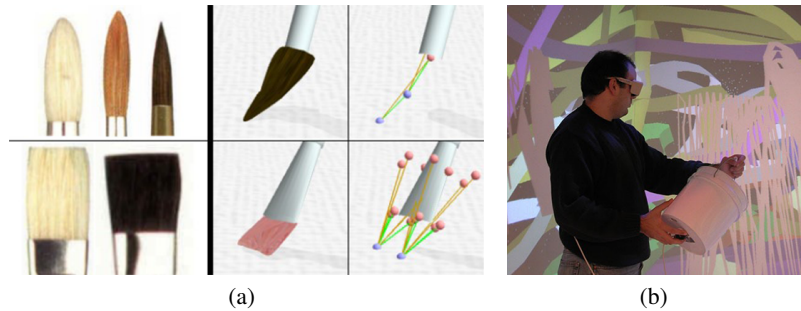


Figure 8.17: Examples of systems that use physical-tools. (a) DAB [78] © ACM and (b) CavePainting [19]

Second, the Heat Gun melts the existing materials as if it was Styrofoam. And the Sandpaper smooths the shape by wearing away the ridges and filling the valleys. Finally, DAB [78] emulates the feeling of drawing with different brushes. Their 3D haptic brush models also affect the stroke characteristic.

In this section, we presented six stroke-creation interaction techniques with specific advantages and disadvantages. When designing a 3D sketching user interface, designers need to consider which stroke control interaction is the best for their current goal. For example, if users are going to draw objects that consist primarily of straight lines or of curves. However, it is important to remember that users should be able to use the same sketching tool for different situations. As Shneiderman wrote, “design with low thresholds, high ceilings, and wide walls” [79]. In other words, a 3D sketching user interface should be easy to understand, while providing a wide range of functionalities.

8.2.1.2 3D Drawing Tools

In this section, we discuss the functional affordances of the interaction techniques that help users draw more accurately, while keeping the fluidity and ease of sketching. These interaction techniques, also called drawing-tools, try to diminish the ergonomic and technological issues of 3D sketching by providing additional drawing tools inside the virtual environment that solve some of the problems mentioned in Section 6.2.2.1. The drawing-tools available in a user interface depend on the type of sketch the interface produces. The type of user also affects the drawing tools available. Finally, to avoid confusing users, it is important to consider

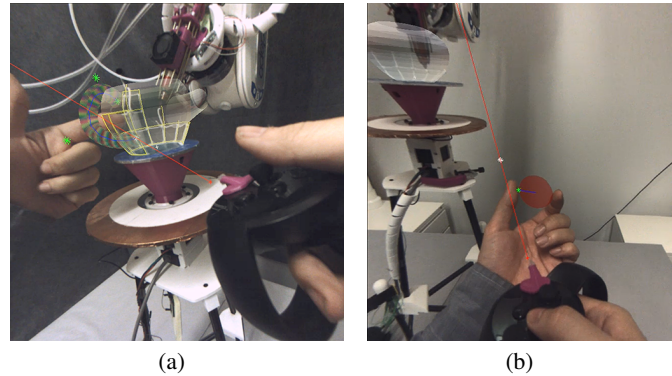


Figure 8.18: RoMA [38] uses rays to point

the number and type of drawing-tools a user interface provides, as some interaction techniques try to solve the same problem in different ways. Next, we will discuss some popular drawing tools and discuss in which scenarios they are useful.

Rays

Using a ray to show where the user controller intersects with the virtual environment is widely used in virtual reality and augmented reality applications. The goal of the rays interaction technique is to help users know where they are pointing. The same principle applies to 3D sketching interfaces, where a ray shows the exact intersection position of the drawing device with the drawn sketch before they touch it.

The rays interaction technique is useful for editing sketches. For example, in RoMA [38] a ray shows users the part of the sketch they are adjusting as shown in Figure 8.18. In the more modelling-oriented user interfaces, rays also helps users identify the selected edge, vertex or surface. Finally, the rays interaction technique can work together with shape guides and drawing planes to show the position where the stroke is going to be drawn.

Shape-guides

The shape-guides interaction technique consists of having guides inside the virtual environment. These guides help users draw more accurately

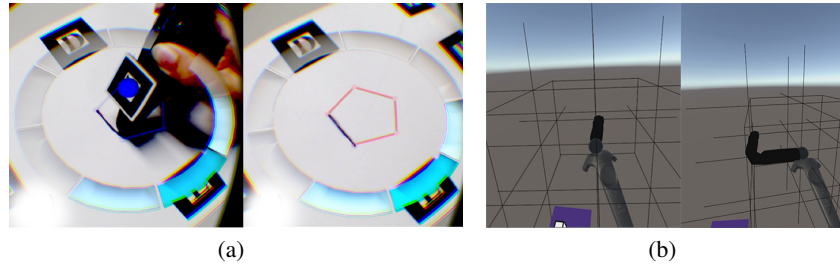


Figure 8.19: The shape-guides interaction technique provides users with guides inside the virtual environment. For example, (a) WireDraw [80] and (b) Smart3DGuides [14].

by allowing them to visualise the errors they make. Most shape guides interaction techniques react to the user's actions, but this is not a requirement. Designers of shape-guide interaction techniques need to consider the shape and functionality of the guide, as these characteristics change depending on the type of sketch the user interface creates.

One shape-guide interaction technique helps users visualise their next stroke [50, 80] by displaying guides a user can trace over. For example, WireDraw [80], shown in Figure 8.19(a), helps users create physical models with a 3D extruder pen. WireDraw needs to know the drawn object so it can divide it into strokes, and order the strokes into steps. After that, while the user is drawing, the system shows the position and shape of the next stroke users need to draw. Another shape-guide interaction technique uses a physical object as a guide. For example, Milosevic et al.'s system uses a pen to trace over objects that the system automatically translates into 3D models. These shape-guides interaction techniques are useful in situations where the user already knows what they are drawing and wants to transfer the model into another medium. For example, WireDraw requires a digital model to create the physical object. Another use case is when the user has a model and wants to alter it or mix parts of different models. Here using 3D sketching over 3D modelling tools might be beneficial thanks to the properties we have already mentioned, such as fluidity and expressiveness among others. However, it is important to note that tracing virtual objects can make the user's stroke less accurate [24].

Other shape-guide interaction techniques help users orient their drawings in space [14, 28, 27]. For example, in Kim et al.'s system the user

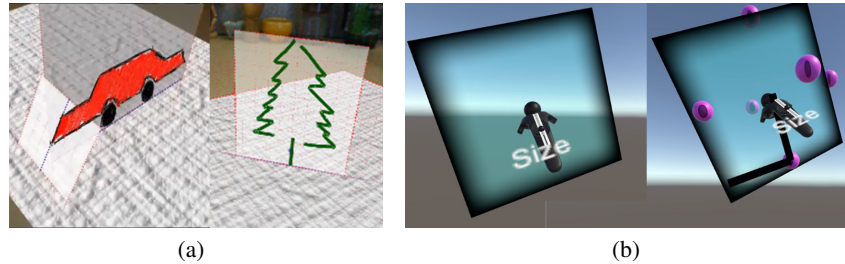


Figure 8.20: Drawing-planes give users surfaces in space to draw on. For example, (a) Napkin sketch [83, 84] for AR and (b) Multiplanes [64] for VR.

creates scaffolds by waving their hands. These scaffolds define the size and general shape of the drawn sketch. Another example is Smart3DGuides [14], shown in Figure 8.19(b), that has local and global reference frames that help users keep the shape-likeness of their sketches consistent. Finally, in SketchingWithHands [28] users use virtual models of their hands in different poses to help them sketch objects that fit in the user's hand. Using shape-guides interaction techniques to help users visualise the final shape assists them in planning their next stroke [14]. However, it is important to avoid filling the virtual environment with guides, as this can be counterproductive, because it can distract or confuse users [82].

Drawing-planes

In Section 6.2, we discussed that one common error users make when 3D sketching is bending the stroke in depth. The drawing-planes interaction technique helps users prevent this error by projecting the stroke to a plane [64, 15, 28, 27, 29, 38, 83, 84]. In other words, this interaction technique gives users a virtual canvas on which they can draw.

The drawing-planes interaction technique has different implementations depending on the device used to display the virtual environment. In AR headsets, the drawing-planes interaction technique helps extend the real-world to mid-air. For example, Napkin sketch [83], shown in Figure 8.20(a), uses the real-world to create a base plane. Users can define new canvas by drawing a line on the base-plane to set the orientation of the new plane. Then, the system uses the base-plane normal to create the new plane. Finally, users can further position the new plane by rotating it. In VR headsets, the drawing-planes interaction technique shows a canvas

inside the virtual environment. For example, Multiplanes [64], shown in Figure 8.20(b), automatically generates drawing planes using the user's hand orientation and previously created planes. Multiplanes uses rules to create parallel, perpendicular, acute angles, and free orientation planes. Another VR example is Hyve-3D [15], where users draw in a tablet that they directly position in the virtual environment. The tablet works as the drawing plane on to which the system snaps the stroke.

Finally, the drawing-plane interaction technique is useful for interfaces that use a 2D screen to display the virtual environment [28, 27, 29]. In these devices, users position the plane in the virtual environment using common interactions as in 3D CAD systems, such as widgets. Then, the system projects the 2D strokes into these planes, which allows users to create 3D sketches. For example, users translate the drawing-plane using a drag and drop interaction in Mohanty et al.'s system. Users can also define a pivot and a rotation radius to rotate the drawing-plane.

Drawing-surfaces

Like drawing-planes, the drawing-surfaces interaction technique [11, 50, 27, 83, 84] helps users improve their strokes. The difference between both interaction techniques is that the drawing-surfaces interaction technique is not limited to planar surfaces. Instead, users can draw on curved surfaces, including the 3D model of an object. The advantage of drawing-surfaces over drawing-planes is that users have more freedom to create complex shapes.

One example is SymbiosisSketch [11], shown in Figure 8.21(a), where users can draw on previously created strokes. SymbiosisSketch also allows users to create drawing planes and surfaces. Another example of this interaction technique is Kim et al.'s system, shown in Figure 8.21(b), where users draw in scaffolds created by their hand gestures.

In this section, we discussed four different interaction techniques that help users draw more accurately. However, this is not a comprehensive list, as designing drawing-tools that are easy to use and learn is still an open area of research. As with the stroke creation interaction techniques, the best drawing-tool is dependent on the sketch the user is drawing. This decision will also depend on the device used to create the 3D sketch, as the drawing-tool should enhance their affordances.

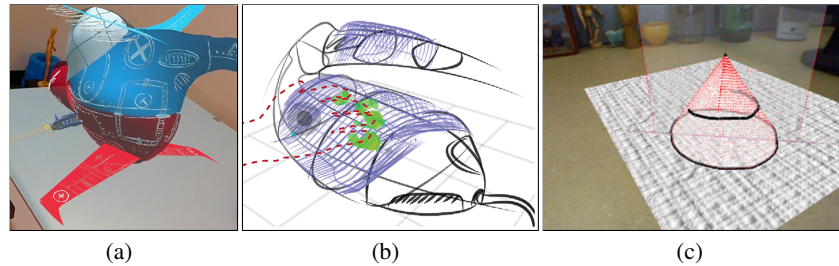


Figure 8.21: Drawing-surfaces give users surfaces of different shapes to draw on. For example, (a) SymbiosisSketch [11], (b) [27]’s system, and (c) Napkin sketch [83, 84].

8.2.2 Physical affordances

Physical affordances are the characteristics of an interaction device that helps users achieve their tasks. For example, a button needs to have an adequate size and an easy-to-access location [72]. Designers of user interfaces for 3D sketching need to consider the physical affordances of the input device used to create strokes. It is also important to consider if the user interface is going to provide haptic feedback to the users and the physical affordances this feedback provides. In the following sections, we are going to discuss these two artefacts.

8.2.2.1 Input Device

The pen or pencil a person uses to draw gives different properties to the sketch. For example, the nib used in a calligraphy pen affects the width of the font, and the graphic harness of a pencil tells how hard or soft the stroke is going to be. For 3D sketching user interfaces, the ergonomics of the input device affects user performance [85, 86]. Some of the physical affordances that designers of 3D user interfaces need to consider are the grip type [87], and the weight distribution [88]. Next, we are going to discuss different input devices used for 3D sketching and which affordances they provide.

Controllers

The last wave of **VR** and **AR** headsets such as Oculus Rift S [89] and Vive Pro [90] come with controllers that work for a wide range of applications.

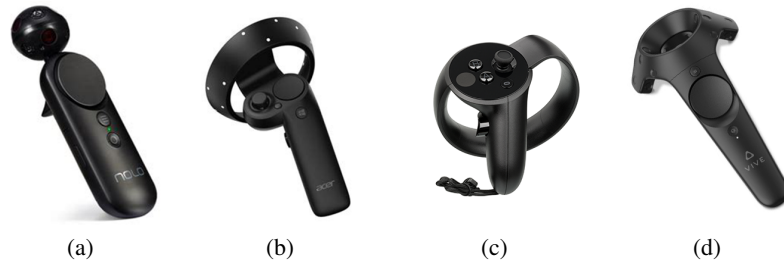


Figure 8.22: Current controllers for VR. (a) [92], (b) Windows Mixed Reality Headset [93], (c) Oculus [89], and (d) [90]

See Figure 8.22. For a 3D sketching user interface, these controllers are easy to use, as they are already integrated with the headset [14, 91, 22, 38, 45]. However, most of these controllers do not provide the physical affordances that a user needs to create accurate sketches. For example, most controllers use a power grip that helps users press buttons, but it makes doing precise movements difficult [85]. Designers of user interfaces for 3D sketching can solve the controllers' accuracy issues using functional affordances. However, usually, it is better to use other input devices.

Pens

Most people learn how to use a pen to write and sketch in their infancy, and they can translate this knowledge into 3D sketching. The design of the pen-like device encourages users to hold the pen using their fingers. Zhai et al. [86] found that using the finger muscles to grip the input device has better performance than using the wrist or elbows muscles. For example, users can grab the pen using the precision grip, where users hold the pen with their thumb and index fingers, which gives precise control of the movement [87, 94]. Thanks to this affordance many user interfaces for 3D sketching use a pen as an input device [95, 51, 27, 81, 53, 26, 96, 97, 80, 98].

When designing a pen for a 3D sketching system, it is important to consider the way the system tracks the pen. Some devices use markers [96, 97]. For example, ARPen [96], shown in Figure 8.23(a) uses a computer vision algorithm to track pen movement in mid-air. Then using a smartphone users can visualise the drawn sketch. Other user interfaces track the pen using a camera. For example, [81]'s system uses IR-LEDs inside the pen that a camera tracks.

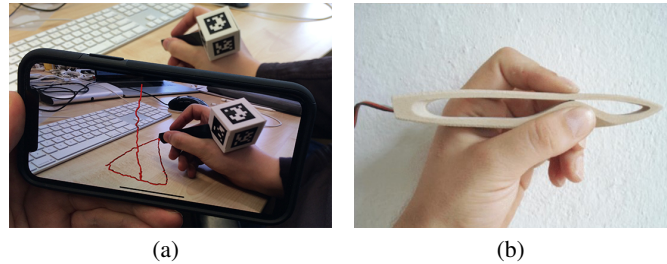


Figure 8.23: Pens are common tools used to sketch. For 3D sketching, it is important to consider how the system tracks the pen position in space. Examples of pens for 3D sketching. (a) ARPen [96] and (b) [8, 99]'s pen.

Finally, pens can also provide other physical affordances that enhance the user experience like haptic feedback [95, 8, 98], and force-sensitive sensors [8]. For example, Israel et al.'s pen (Figure 8.23b) allows users to draw by pressing the upper component of the pen until it touches the lower component, and depending on the force by which the user pressed the pen the width of the stroke changes from 1 to 8 mm.

Gestures

Gestures are input events using human hands, and they consist of the movement of fingers and arms in mid-air. One advantage of interactions with gestures is that gesture-based interfaces can reduce the complexity of interaction between humans and computers [100]. And a specific advantage of using gestures for 3D sketching is that this interaction allows for detailed control using the index finger, and more dynamic control using the whole arm as shown by [101]. See Figure 8.24.

However, when designing a user interface for 3D sketching, it is important to distinguish between manipulative and communicative gestures [102]. Manipulative gestures are those used to interact with objects, for example, take an object, move it somewhere and release it. These gestures do not require attention resources, as using hands is a natural way to interact with the physical world and objects in it, and therefore do not disturb the sketching process. On the other hand, communicative gestures use gestures as symbolic shapes that communicate specific acts. For example, the [American sign language \(ASL\)](#). When using communicative gestures for 3D sketching, users need to remember the meaning behind each sign, which can break



Figure 8.24: Gestures allow users to draw strokes using their fingers and arms. For example, (a) users draw with their fingers in AiRSculpt [33], and (b) with their arms in Surface drawing [101].

the sketching process [41]. Another problem to consider is that current gesture recognition technology is not accurate enough to make the interaction seamless. For example, users can only move their hands after the system detects their gestures. Therefore, even using manipulative gestures can be clumsy, especially as there needs to be a second gesture or input device to tell the system when to start and end a stroke. Finally, gestures can also fatigue the user, known as the gorilla arm effect [103].

For 3D sketching, the gestures interaction technique can use a single hand to interact with the sketch [16, 104, 101, 66]. Usually, these systems use the index finger to draw strokes, for example, Dudley et al.'s system and Virtual Hand [104]. This interaction technique can also use all the fingers to create strokes [44] or the palm [101, 66]. Finally, gestures can use two hands to control the sketch. In these user interfaces, users draw with one hand and edit the sketch using both hands [105, 66, 33]. When evaluating gesture interfaces, previous work found that they present the accuracy problems of 3D sketching [16, 33]. Some users also have problems doing the appropriate gesture to activate an option, which affects the usability of the interaction technique [33].

Finally, there are different ways to track the user's hands and detect their gestures. One way is to use a camera [16, 105, 66, 33], another way is to use a glove [44, 104, 101].

Other-inputs

Other projects have used interesting interaction devices to create strokes that do not emulate using a pen to draw. Here we call them other-inputs.

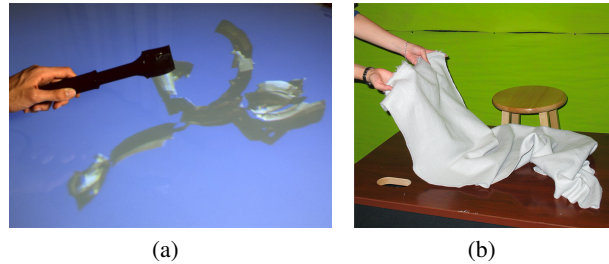


Figure 8.25: Other-inputs use physical tools to draw strokes. For example, (a) Surface drawing [101] uses a tong, and (b) Leal et al.'s system uses fabric.

Using other-inputs can bring an element of exploration and creativity to the design process, as the user interface can become part of the process. There are also almost endless possibilities to create physical tools that match particular modelling tasks. However, most of the other-inputs are custom made and require technical knowledge of electronics. Finally, users might find it difficult to use these interaction devices if their shape does not correspond with their functionality. One example of other-inputs is the bucket used to splash paint in CavePainting [19]. Other examples are the tongs and magnet used to edit the sketch in Surface Drawing [101] as shown in Figure 8.25(a). Finally, Leal et al.'s system, shown in Figure 8.25(b) uses fabric to create strokes. This material is flexible to manipulate but strong enough to keep the shape.

The new generation of high-quality VR and AR headsets that have their own controllers made developing 3D sketching user interfaces simpler. Especially as designers do not need to think about the hardware elements. However, in this section, we discussed some of the problems that the current generation of commercial controllers and those made and used in research have. We also presented three other input devices that have their own set of affordances, which help users sketch more accurately. Or in the case of other-inputs, these devices also help user's creativity. We encourage designers of 3D sketching user interfaces to consider the type of controllers their interface uses.

8.2.2.2 Haptic feedback

When interacting with objects, humans get a tactile sensation that helps us distinguish different properties of the object, which is known as haptic

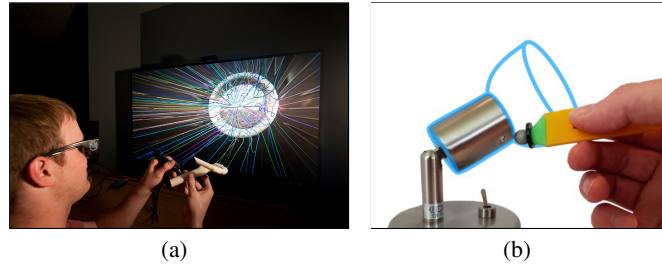


Figure 8.26: Drawing over real-objects allows users to experience haptic feedback when sketching over them. For example, (a) [110]’s system and (b) [35]’s study.

feedback. However, this sensation is not present when interacting with virtual objects unless we use a device capable of simulating the feeling of touching an object by applying forces, vibrations, or motions [106]. For 3D sketching, haptic feedback helps users improve their accuracy [19, 107]. Next, we discuss different interaction devices and techniques that recreate the feeling of haptic feedback. Each of these interaction devices and techniques has its own set of affordances and problems.

Real-objects

Real-objects provide passive haptic feedback when the user touches them, and this physical materiality can affect the user sketch. For example, using a pen to draw provides haptic feedback that affects the way users hold and use the pen to draw. For example, the pen-shaft shape and diameter affect the way users control their movement while sketching [88]. Also, a lightweight pen prevents user fatigue when using the precision grip [85, 108]. Finally, it is this physical materiality that allows users to guide the pen precisely in a precision grip, since slight movements of the fingertips, which are transferred to the pen, resulting in minimal changes in the position of the pen, which in turn are perceived by the fingertips [109].

It is also possible to use physical 3D models of shapes that users can draw over [110, 81]. For example, Jackson and Keefe [110] use 3D printed shapes of scientific datasets that users can explore by sketching over them as illustrated in Figure 8.26. Using a 3D model as haptic feedback can bring some inaccuracies to the stroke, especially if the tip of the device does not fit

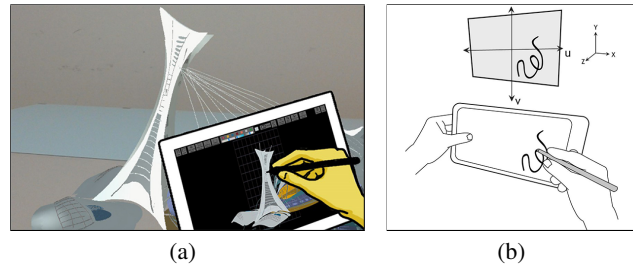


Figure 8.27: Examples of systems that use portable surfaces. (a) SymbiosisSketch [11] and (b) Hyve-3D [15]

in at the edge of a concave or convex shape, as users cannot correctly trace over the outline of the object [35].

Portable surfaces

Portable-surfaces use touch devices to give users a physical surface they can touch. Some user interfaces use a mobile device [56, 11, 15, 83, 18, 111, 112], and other user interfaces use a tablet without a screen [58, 26]. Portable-surfaces allow users to utilise their 2D drawing skills to sketch in 3D by using the screen as a physical canvas. To position the stroke in place, users point the mobile device to the selected object. For example, SymbiosisSketch [11] uses a tablet to add details to previously drawn objects as shown in Figure 8.27(a). Users need to point the tablet at the object they want to draw on. Another approach is to use the tablet as an interaction device, where users also draw. For example, Hyve-3D [15] uses the tablet as a 3D cursor and as a drawing surface as shown in Figure 8.27(b). Users move the canvas by using touch gestures and draw on the canvas using one finger. In SketchTab3D [112] users 2D sketches using the tablet screen. Then, users position these sketches in space using the tablet. Once a sketch is attached to a tablet, users can continue updating this sketch even if they are not directly looking at it.

One problem with portable-surfaces is that users can not feel the shape of the drawn object, or its material. Another problem is that in some examples, users need to keep the tablet static while drawing with one hand, which can be tiring. However, one way to avoid this problem is to allow users to move the tablet without moving the sketch, for example SketchTab3d [112].

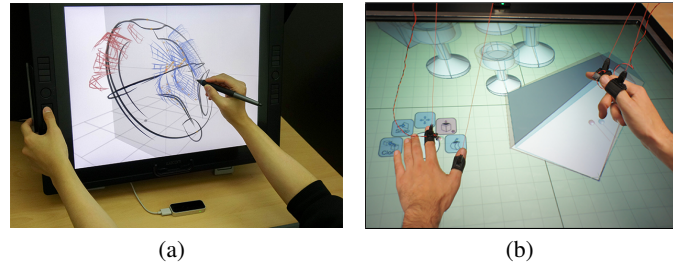


Figure 8.28: Examples of systems that use fixed surfaces. (a) [27]'s system and (b) Mockup Builder [54]

Fixed surfaces

Fixed-surfaces are tables or screens that have touch capabilities, as illustrated in Figure 8.28. Users draw on the screen by using their fingers or other objects. This characteristic makes the fixed-surfaces devices have similar affordances to the portable-surface devices. One advantage of fixed-surfaces over portable-surfaces is that the user does not carry the device, which prevents fatigue.

However, it is important to consider how to change the user's viewpoint when using a fixed-surface device. Some user interfaces use 3D navigation techniques to change the user's viewpoint position [27, 28]. These implementations can be useful to expert designers that have a lot of experience working with 3D CAD systems, as they know how to manipulate the camera fast. However, more novice users can find this approach difficult to use [113]. Other devices use AR headsets to see the shapes outside the screen [54, 55, 57]. These user interfaces are useful for collaborative environments, as many users can see the drawn shapes from multiple viewpoints. Also, most touch tables can support multiple inputs, which helps different designers draw at the same time.

Force feedback

Force-feedback devices use different technologies to manipulate the movement of the device held by the user, see Figure 8.29. For example, vibrotactile actuators [114], pneumatic actuators [115], electrical muscle stimulation (EMS) [116] and mechanical actuators [117]. The way these devices work changes depending on the technology used and a designer of

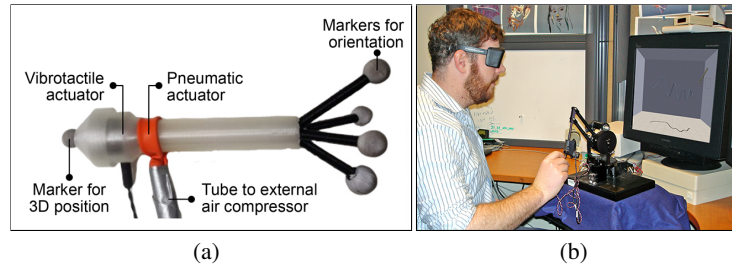


Figure 8.29: Examples of systems that use force-feedback devices. (a) VRSketchPen [98] and (b) Dynamic Dragging [75]

a user interface for 3D sketching needs to choose the correct technology for their system. For example, some devices are fixed in a position, such as The Touch [118] or Phantom [119] among others, while others allow users to move, such as VRSketchPen [98] among others.

Force-feedback devices are intended to feel the size, shape and properties of different objects. For example, feel the difference when modelling with clay versus wood, or feel the edges of a virtual object. One way to use a force-feedback device is to show the user the position of the drawing canvas [29] or the modelled shape [31]. Another way is to snap the controller to the stroke. For example, Drawing on Air [74] and Dynamic Dragging [75] use 3D tape drawing, and utilise a force-feedback device to constrain the stylus tip to remain on the line segment connecting the two hands. Dynasculpt [20] uses the force-feedback device to alter the sculptural qualities of the stroke. Finally, VRSketchPen [98] allows users to feel the position and the texture of the touched object.

One lesson to take away from this section is that haptic feedback is an important component of any sketching interface. Even the passive haptic feedback of pen-like devices to sketch in VR will affect how the user sketches. When using different types of haptic feedback, it is important to consider how they affect the sketching action. In this section, we also presented other types of passive haptic feedback, like touch surfaces that allow users to utilise their 2D drawing skills to sketch in 3D. Finally, there are also force-feedback devices that create the sensation of touching a virtual object.

8.3 Summary

In this chapter, we present different interaction devices or techniques that help users sketch in 3D and discuss the various advantages and disadvantages of each project. We classified these interaction devices or techniques by the type of sketch they aim to create (conceptual, technical or a mix). We also discuss the functional and physical affordances each project presents. Our goal was to help designers of future interaction devices or techniques for 3D sketching a general overview of the area so they can design their own interaction devices or techniques.

References

- [1] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, and K. Akeley. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1996.
- [2] C. K. Lim. An Insight Into The Freedom Of Using A Pen: Pen-based System And Pen-and-Paper. In *Proceedings of the Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA '03)*, pages 385–393, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.759>.
- [3] A. T. Purcell and J. S. Gero. Drawings and the Design Process. *Design Studies*, 19(4):389–430, 1998. ISSN 0142694X. doi: 10.1016/S0142-694X(98)00015-5. URL <http://linkinghub.elsevier.com/retrieve/pii/S0142694X98000155>.
- [4] D. G. Ullman, S. Wood, and D. Craig. The Importance of Drawing in the Mechanical Design Process. *Computers and Graphics*, 14(2):263–274, 1990. ISSN 00978493. doi: 10.1016/0097-8493(90)90037-X.
- [5] H. R. Hartson. Cognitive, Physical, Sensory, And Functional Affordances In Interaction Design. *Behaviour and Information Technology*, 22(5):315–338, 2003. ISSN 0144929X. doi: 10.1080/01449290310001592587.
- [6] D. A. Norman. *The Design of Everyday Things*. Basic Books, Inc., USA, 2002. ISBN 9780465067107.
- [7] C. L. Dym. *Engineering Design: A Synthesis of Views*. Cambridge University Press, 1994.
- [8] J. H. Israel, E. Wiese, M. Mateescu, C. Zöllner, and R. Stark. Investigating Three-dimensional Sketching For Early Conceptual Design—results From Expert Discussions And User Studies. *Computers & Graphics*, 33(4):462 – 473, 2009. ISSN 0097-8493. doi: <https://doi.org/10.1016/j.cag.2009.05.005>. URL <http://www.sciencedirect.com/science/article/pii/S0097849309000855>.

- [9] J. H. Israel, J. Hurtienne, A. Pohlmeier, C. Mohs, M. Kindsmüller, and A. Naumann. On Intuitive Use, Physicality And Tangible User Interfaces. *International Journal of Arts and Technology*, 2:348–366, 03 2009. doi: 10.1504/IJART.2009.02924.
- [10] I. M. Verstijnen, J. M. Hennessey, C. van Leeuwen, R. Hamel, and G. Goldschmidt. Sketching and Creative Discovery. *Design Studies*, 19(4):519–546, oct 1998. ISSN 0142694X. doi: 10.1016/S0142-694X(98)00017-9. URL <http://linkinghub.elsevier.com/retrieve/pii/S0142694X98000179>.
- [11] R. Arora, R. Habib K., T. Grossman, G. Fitzmaurice, and K. Singh. Symbiosissketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2018.
- [12] B. Yee, Y. Ning, and H. Lipson. Augmented Reality In-Situ 3D Sketching of Physical Objects. *Proceedings of the Intelligent UI Workshop on Sketch Recognition*, pages 1–4, 2009. URL http://creativemachines.cornell.edu/papers/TUI09f_Yee.pdf.
- [13] G. Wesche and H. P. Seidel. FreeDrawer: A Free-Form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '01)*, page 167, New York, New York, USA, 2001. ACM Press. ISBN 1581134274. doi: 10.1145/505008.505041. URL <http://portal.acm.org/citation.cfm?doid=505008.505041>.
- [14] M. D. Barrera Machuca, W. Stuerzlinger, and P. Asente. Smart3DGuides: Making Unconstrained Immersive 3D Drawing More Accurate. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–13, New York, NY, USA, nov 2019. ACM. ISBN 9781450370011. doi: 10.1145/3359996.3364254. URL <http://dl.acm.org/doi/10.1145/3359996.3364254>.
- [15] T. Dorta, G. Kinayoglu, and M. Hoffmann. Hyve-3D and the 3D Cursor: Architectural co-design with freedom in Virtual Reality. *International Journal of Architectural Computing*, 14(2):87–102, 2016.
- [16] J. J. Dudley, H. Schuff, and P. O. Kristensson. Bare-Handed 3D Drawing in Augmented Reality. In *Proceedings of the ACM Conference on Designing Interactive Systems (DIS '18)*, pages 241–252, 2018. ISBN 9781450351980. doi: 10.1145/3196709.3196737. URL <http://dl.acm.org/citation.cfm?doid=3196709.3196737>.
- [17] S. Eroglu, S. Gebhardt, P. Schmitz, D. Rausch, and Torsten W. K. Fluid Sketching—Immersive Sketching Based on Fluid Flow. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 475–482, Reutlingen, March 2018. IEEE. ISBN 978-1-5386-3365-6. doi: 10.1109/VR.2018.8446595. URL <https://ieeexplore.ieee.org/document/8446595/>.
- [18] T. Fleisch, G. Brunetti, P. Santos, and A. Stork. Stroke-input Methods For Immersive Styling Environments. *Proceedings - Shape Modeling International SMI 2004*, pages 275–283, 2004. doi: 10.1109/SMI.2004.1314514.

- [19] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola. CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pages 85–93, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581132921. doi: 10.1145/364338.364370. URL <https://doi.org/10.1145/364338.364370>.
- [20] S. Snibbe, S. Anderson, and B. Verplank. Springs and Constraints for 3D Drawing. In *Proceedings of the Third Phantom Users Group*, 1998.
- [21] S. Tano, N. Kanayama, T. Hashiyama, J. Ichino, and M. Iwata. 3D Sketch System Based On Life-sized And Operable Concept Enhanced By Three Design Spaces. *Proceedings - 2014 IEEE International Symposium on Multimedia, ISM 2014*, pages 245–250, 2015. doi: 10.1109/ISM.2014.54.
- [22] T. McGraw, E. Garcia, and D. Sumner. Interactive Swept Surface Modeling In Virtual Reality With Motion-tracked Controllers. In *Proceedings of the EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (SBIM '17)*, pages 1–9, New York, New York, USA, 2017. ACM Press. ISBN 9781450350792. doi: 10.1145/3092907.3092908. URL <http://dl.acm.org/citation.cfm?doid=3092907.3092908>.
- [23] Y. Kim, B. Kim, J. Kim, and Y. J. Kim. CanvoX: High-resolution VR Painting in Large Volumetric Canvas, 2017.
- [24] R. Arora, R. H. Kazi, F. Anderson, T. Grossman, K. Singh, and G. Fitzmaurice. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 5643–5654, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3025474. URL <https://doi.org/10.1145/3025453.3025474>.
- [25] M. D. Barrera Machuca, W. Stuerzlinger, and Paul Asente. The Effect of Spatial Ability on Immersive 3D Drawing. In *Proceedings of the 2019 on Creativity and Cognition*, C&C '19, page 173–186, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359177. doi: 10.1145/3325480.3325489. URL <https://doi.org/10.1145/3325480.3325489>.
- [26] M. A. Schroering, C. M. Grimm, and R. Pless. A New Input Device for 3D Sketching. *Vision Interface*, pages 311–318, 2003.
- [27] Y. Kim, S. G. An, J. H. Lee, and S. H. Bae. Agile 3D Sketching with Air Scaffolding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '18)*, pages 1–12, 2018. ISBN 9781450356206. doi: 10.1145/3173574.3173812. URL <http://dl.acm.org/citation.cfm?doid=3173574.3173812>.
- [28] Y. Kim and S. H. Bae. SketchingWithHands: 3D Sketching Handheld Products with First-Person Hand Posture. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 797–808, 2016.

- [29] R. R. Mohanty, Umema H. B., E. Ragan, and Vinayak. Kinesthetically Augmented Mid-Air Sketching of Multi-Planar 3D Curve-Soups. In *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME 2018)*, pages 1–11, 2018.
- [30] D. Donath and H. Regenbrecht. Using Virtual Reality Aided Design Techniques For Three-dimensional Architectural Sketching. In *Design Computation, Collaboration, Reasoning, Pedagogy. ACADIA Conference Proceedings*, pages 201–212, 1996.
- [31] T. A. Galyean and J. F. Hughes. Sculpting: An Interactive Volumetric Modeling Technique. *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991*, 25:267–274, 1991. doi: 10.1145/122718.122747.
- [32] A. Leal, L. Schaefer, D. Bowman, F. Quek, and C. Stiles. 3D Sketching Using Interactive Fabric For Tangible And Bimanual Input. *Proceedings - Graphics Interface*, pages 49–56, 2011. ISSN 07135424.
- [33] S. A. Jang, H. Kim, W. Woo, and G. Wakefield. AiRSculpt: A Wearable Augmented Reality 3D Sculpting System. In *Distributed, Ambient, and Pervasive Interactions, Lecture Notes in Computer Science*, volume 8530, pages 130–141. Springer International Publishing, 2014. ISBN 978-3-319-07787-1. doi: 10.1007/978-3-319-07788-8_13.
- [34] F. L. Krause and J. Lüddemann. *Virtual Clay Modelling*, pages 162–175. Springer US, Boston, MA, 1997. ISBN 978-0-387-35187-2. doi: 10.1007/978-0-387-35187-2_14. URL https://doi.org/10.1007/978-0-387-35187-2_14.
- [35] P. Wacker, A. Wagner, S. Voelker, and J. Borchers. Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality. In *Proceedings of the Symposium on Spatial User Interaction, SUI '18*, page 25–35, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357081. doi: 10.1145/3267782.3267788. URL <https://doi.org/10.1145/3267782.3267788>.
- [36] Autodesk. Autocad. <https://www.autodesk.com/products/autocad/overview>, 2020.
- [37] Dassault Systems. Solidworks. <https://www.solidworks.com/>, 2020.
- [38] H. Peng, J. Briggs, C. Y. Wang, K. Guo, J. Kider, S. Mueller, P. Baudisch, and F. Guimbretière. Roma: Interactive fabrication with augmented reality and a Robotic 3D printer. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April:1–12, 2018. doi: 10.1145/3173574.3174153.
- [39] M. F. Deering. The HoloSketch VR Sketching System. *Communications of the ACM*, 39(5):54–61, 1996. ISSN 00010782. doi: 10.1145/229459.229466. URL <http://portal.acm.org/citation.cfm?doid=229459.229466>.

- [40] F. Bruno, M. L. Luchi, M. Muzzupappa, and S. Rizzuti. A Virtual Reality Desktop Configuration for Free-Form Surface Sketching. In *Proceedings on XIV Congreso Internacional de Ingeniería Gráfica*, 2002.
- [41] J. Petruschat. Some Remarks on Drawing. *Form+zweck. How to Handle Hands?*, 18: 70–77, 2001.
- [42] S. Nam and Y. Chai. SPACESKETCH: Shape Modeling With 3D Meshes And Control Curves In Stereoscopic Environments. *Computers and Graphics (Pergamon)*, 36(5): 526–533, 2012. ISSN 00978493. doi: 10.1016/j.cag.2012.03.012. URL <http://dx.doi.org/10.1016/j.cag.2012.03.012>.
- [43] J. H. Kwon, H. W. Choi, J. I. Lee, and Y. H. Chai. Free-hand stroke based NURBS surface for sketching and deforming 3D contents. In *Advances in Multimedia Information Processing - PCM 2005*, volume 3767 LNCS, pages 315–326, 2005. ISBN 3540300279. doi: 10.1007/11581772_28.
- [44] M. Fuge, M. E. Yumer, G. Orbay, and L. B. Kara. Conceptual Design And Modification Of Freeform Surfaces Using Dual Shape Representations In Augmented Reality Environments. *CAD Computer Aided Design*, 44(10):1020–1032, 2012. ISSN 00104485. doi: 10.1016/j.cad.2011.05.009. URL <http://dx.doi.org/10.1016/j.cad.2011.05.009>.
- [45] X. Zhu, L. Song, L. You, M. Zhu, X. Wang, and X. Jin. Brush2Model: Convolution Surface-Based Brushes for 3D Modelling in Head-Mounted Display-Based Virtual Environments. *Computer Animation and Virtual Worlds*, 28(3-4):1–10, 2017. ISSN 1546427X. doi: 10.1002/cav.1764.
- [46] Fabio Bruno, Maria Laura Luchi, Sergio Rizzuti, and Università Della Calabria. The Over-sketching Technique For Free-hand Shape Modelling In Virtual Reality. In *In Proceedings of Virtual Concept 2003, Biarritz*, pages 5–7, 2003.
- [47] R. Schmidt, A. Khan, G. Kurtenbach, and K. Singh. On Expert Performance in 3D Curve-Drawing Tasks. In *Proceedings of the EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (SBIM '09)*, pages 133–140, New York, New York, USA, 2009. ACM Press. ISBN 9781605586021. doi: 10.1145/1572741.1572765. URL <http://dl.acm.org/citation.cfm?id=1572741.1572765><http://portal.acm.org/citation.cfm?doid=1572741.1572765>.
- [48] S. Lee and J. Yan. The Impact Of 3D Cad Interfaces On User Ideation: A Comparative Analysis Using Sketchup And Silhouette Modeler. *Design Studies*, 44:52–73, 2016. ISSN 0142694X. doi: 10.1016/j.destud.2016.02.001. URL <http://dx.doi.org/10.1016/j.destud.2016.02.001>.
- [49] D. Veisz, E. Z. Namouz, S. Joshi, and J. D. Summers. Computer-aided Design Versus Sketching: An Exploratory Case Study. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 26(3):317–335, 2012. ISSN 08900604. doi: 10.1017/S0890060412000170.

- [50] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. Creating Principal 3d Curves With Digital Tape Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, pages 121–128, New York, New York, USA, 2002. ACM Press. ISBN 1581134533. doi: 10.1145/503376.503398. URL <http://portal.acm.org/citation.cfm?doid=503376.503398>.
- [51] B. Jackson and D. F. Keefe. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1442–1451, 2016.
- [52] R. De Amicis, F. Bruno, A. Stork, and M. L. Luchi. The Eraser Pen: A New Interaction Paradigm for Curve Sketching in 3D. *DS 30: Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik*, pages 465–470, 2002. URL <https://www.designsociety.org/publication/29603/TheEraser+Pen{+}3A+A+New+Interaction+Paradigm+for+Curve+Sketching+in+3D>.
- [53] H. Perkunder, J. H. Israel, and M. Alexa. Shape Modeling with Sketched Feature Lines in Immersive 3D Environments. In *Proceedings of the EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (SBIM '10)*, pages 127–134, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association. ISBN 978-3-905674-25-5.
- [54] B. R. De Araújo, G. Casiez, and J. A. Jorge. Mockup Builder: Direct 3D Modeling On and Above the Surface in a Continuous Interaction Space. In *Proceedings of the Graphics Interface Conference (GI '12)*, pages 173–180, 2012. ISBN 978-1-4503-1420-6. URL <http://dl.acm.org/citation.cfm?id=2305305>.
- [55] B. R. De Araújo, G. Casiez, J. A. Jorge, and M. Hachet. Mockup Builder: 3D modeling On and Above the Surface. *Computers & Graphics*, 37(3):165 – 178, 2013. ISSN 0097-8493. doi: <https://doi.org/10.1016/j.cag.2012.12.005>. URL <http://www.sciencedirect.com/science/article/pii/S0097849312001811>.
- [56] K. Huo, Vinayak, and K. Ramani. Window-Shaping. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*, pages 37–45, 2017. ISBN 9781450346764. doi: 10.1145/3024969.3024995. URL <http://dl.acm.org/citation.cfm?doid=3024969.3024995>.
- [57] P. Reipschläger and R. Dachselt. Designar: Immersive 3D-modeling Combining Augmented Reality with Interactive Displays. *ISS 2019 - Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pages 29–41, 2019. doi: 10.1145/3343055.3359718.
- [58] H. Seichter. SKETCHAND+ A Collaborative Augmented Reality Sketching Application. *Proceedings of the 8th International Conference on Computer-Aided Architectural Design Research in Asia*, pages 209–222, 2003.
- [59] Trimble. sketchup. <https://www.sketchup.com/>, 2020.

- [60] J. Fišer, P. Asente, and D. Sýkora. ShipShape : A Drawing Beautification Assistant. *Joint Symposium of Computational Aesthetics (CAe) Non-Photorealistic Animation and Rendering (NPAR) Sketch-Based Interfaces and Modeling (SBIM) (Expressive'15)*, 2015. doi: 10.2312/exp.20151178.
- [61] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka. Interactive Beautification: A Technique for Rapid Geometric Design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, page 105–114, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918819. doi: 10.1145/263407.263525. URL <https://doi.org/10.1145/263407.263525>.
- [62] G. Orbay and L. B. Kara. Beautification of Design Sketches Using Trainable Stroke Clustering and Curve Fitting. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):694–708, 2011.
- [63] G. Johnson, M. D. Gross, J. Hong, and E. Y. Do. Computational Support for Sketching in Design: A Review. *Foundations and Trends in Human–Computer Interaction*, 2(1): 1–93, 2009. ISSN 1551-3955. doi: 10.1561/1100000013. URL <http://dx.doi.org/10.1561/1100000013>.
- [64] M. D. Barrera Machuca, P. Asente, W. Stuerzlinger, J. Lu, and B. Kim. Multiplanes: Assisted Freehand VR Sketching. In *Proceedings of the Symposium on Spatial User Interaction*, SUI '18, page 36–47, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357081. doi: 10.1145/3267782.3267786. URL <https://doi.org/10.1145/3267782.3267786>.
- [65] M. Fiorentino, G. Monno, P. A. Renzulli, and A. E. Uva. 3D Sketch Stroke Segmentation and Fitting in Virtual Reality. In *International Conference on the Computer Graphics and Vision*, pages 188–191, 2003. doi: 10.1.1.99.9190.
- [66] S. Shankar and R. Rai. Sketching In Three Dimensions: A Beautification Scheme. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 31(3):376–392, 2017. ISSN 14691760. doi: 10.1017/S0890060416000512.
- [67] Surbhi Gupta, Munish Kumar, and Anupam Garg. Improved object recognition results using SIFT and ORB feature detector. 78(23):34157–34171. ISSN 1573-7721. doi: 10.1007/s11042-019-08232-6. URL <https://doi.org/10.1007/s11042-019-08232-6>.
- [68] Callie Federer, Haoyan Xu, Alona Fyshe, and Joel Zylberberg. Improved object recognition using neural networks trained to mimic the brain's statistical properties. *Neural Networks*, 131:103–114, 2020. ISSN 0893-6080. doi: <https://doi.org/10.1016/j.neunet.2020.07.013>. URL <https://www.sciencedirect.com/science/article/pii/S0893608020302549>.
- [69] D. Cooper. Imagination's Hand: The Role Of Gesture In Design Drawing. *Design Studies*, 54:120–139, 2018. ISSN 0142694X. doi: 10.1016/j.destud.2017.11.001. URL <https://doi.org/10.1016/j.destud.2017.11.001>.

- [70] E. Y. L. Do and M. D. Gross. Drawing as a Means to Design Reasoning. In *Artificial Intelligence in Design '96 Workshop on Visual Representation, Reasoning and Interaction in Design*, pages 1–11, 1996. URL papers2://publication/uuid/02F84179-3333-494B-8588-D4621BCB4CAD.
- [71] W. Buxton. *Sketching User Experiences: Getting The Design Right And The Right Design*. Morgan Kaufmann, San Francisco, 2007.
- [72] H. Rex and S. P. Partha. *The UX Book*. Morgan Kaufmann, Boston, 2012. ISBN 9780465067107.
- [73] Google. Tilt Brush. <https://www.tiltbrush.com/>, 2020.
- [74] D. Keefe, R. Zeleznik, and D. Laidlaw. Drawing on Air: Input Techniques for Controlled 3D line Illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, 2007.
- [75] D. F. Keefe and R. C. Zeleznik. Tech-note: Dynamic Dragging for Input of 3D Trajectories. In *2008 IEEE Symposium on 3D User Interfaces*, pages 51–54, 2008.
- [76] Gravity Sketch. Gravity Sketch. <https://www.gravitysketch.com/>, 2020.
- [77] I. Poupyrev, M. Billinghamurst, S. Weghorst, and T. Ichikawa. The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology, UIST '96*, page 79–80, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917987. doi: 10.1145/237091.237102. URL <https://doi.org/10.1145/237091.237102>.
- [78] B. Baxter, V. Scheib, M. C. Lin, and D. Manocha. DAB: Interactive Haptic Painting with 3D Virtual Brushes. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*, pages 461–468, 2001. doi: 10.1145/383259.383313.
- [79] B. Shneiderman. Creativity Support Tools Accelerating Discovery and Innovation. *Communications of the ACM*, 50(12):20–32, 2007.
- [80] Y. T. Yue, X. Zhang, Y. Yang, G. Ren, Y. K. Choi, and W. Wang. WireDraw. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*, pages 3693–3704, 2017. ISBN 9781450346559. doi: 10.1145/3025453.3025792. URL <http://dl.acm.org/citation.cfm?doid=3025453.3025792>.
- [81] B. Milosevic, F. Bertini, E. Farella, and S. Morigi. A SmartPen for 3D Interaction and Sketch-based Surface Modeling. *International Journal of Advanced Manufacturing Technology*, 84(5-8):1625–1645, 2016. ISSN 14333015. doi: 10.1007/s00170-015-7828-1. URL <http://dx.doi.org/10.1007/s00170-015-7828-1>.
- [82] B. Shneiderman, C. Plaisant, and M. Cohen. *Designing the User Interface*. Pearson, 6th edition, 2016.

- [83] M. Xin, E. Sharlin, and M. Costa Sousa. Napkin Sketch: Handheld Mixed Reality 3D Sketching. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pages 223–226, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939517. doi: 10.1145/1450579.1450627. URL <https://doi.org/10.1145/1450579.1450627>.
- [84] M. Xin. *3D Sketching and Collaborative Design with Napkin Sketch*. PhD thesis, University of Calgary, Canada, 2011. URL <https://utouch.cpsc.ucalgary.ca/docs/NapkinSketch-Thesis-MX.pdf>.
- [85] D. M. Pham and W. Stuerzlinger. Is the Pen Mightier than the Controller? A Comparison of Input Devices for Selection in Virtual and Augmented Reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370011. doi: 10.1145/3359996.3364264. URL <https://doi.org/10.1145/3359996.3364264>.
- [86] S. Zhai, P. Milgram, and W. Buxton. The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 308–315, 1996.
- [87] C. M. Schneck. Comparison of Pencil-Grip Patterns in First Graders with Good and Poor Writing Skills. *The American journal of occupational therapy. : official publication of the American Occupational Therapy Association*, 45(8):701–706, 1991. ISSN 02729490. doi: 10.5014/ajot.45.8.701.
- [88] R. S. Goonetilleke, E. R. Hoffmann, and A. Luximon. Effects Of Pen Design On Drawing And Writing Performance. *Applied Ergonomics*, 40(2):292–301, 2009. ISSN 00036870. doi: 10.1016/j.apergo.2008.04.015.
- [89] Oculus by Facebook. Oculus Rift S: VR Headset for VR Ready PCs. <https://www.oculus.com/rift-s/>, 2019.
- [90] HTC. Vive PRO — The Professional Grade VR Headset. <https://www.vive.com/eu/product/vive-pro/>, 2018.
- [91] H. Gardner, D. Lifeng, Q. Wang, and G. Zhou. Line Drawing In Virtual Reality Using A Game Pad. *{AUIC} '06: {P}roceedings of the 7th {A}ustralasian {U}ser interface conference*, 15:177–180, 2006.
- [92] Nolo. Nolo X1 — 6DoF VR Set. https://www.nolovr.com/nolo_x1, 2020.
- [93] by Acer. Windows Mixed Reality Headset — TAKING YOU TO A NEW REALITY. <https://www.acer.com/ac/en/US/content/series/wmr>, 2020.
- [94] T. H. Falk, C. Tam, H. Schwellnus, and T. Chau. Grip Force Variability And Its Effects On Children's Handwriting Legibility, Form, And Strokes. *Journal of Biomechanical Engineering*, 132(11), 2010. ISSN 01480731. doi: 10.1115/1.4002611.

- [95] M. Fiorentino, A. E. Uva, and G. Monno. The SenStylus: A Novel Rumble-Feedback Pen Device for CAD Application in Virtual Reality. *13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2005, WSCG'2005 - In Co-operation with EUROGRAPHICS, Full Papers*, pages 131–138, 2005.
- [96] P. Wacker, O. Nowak, S. Voelker, and J. Borchers. ARPen : Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '19)*, pages 1–12, 2019. ISBN 9781450359702.
- [97] P. C. Wu, R. Wang, K. Kin, C. Twigg, S. Han, M. H. Yang, and S. Y. Chien. DodecaPen: Accurate 6DoF Tracking of a Passive Stylus. *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 365–374, 2017. doi: 10.1145/3126594.3126664.
- [98] H. Elsayed, M. D. Barrera Machuca, C. Schaarschmidt, K. Marky, F. Müller, J. Riemann, A. Matviienko, M. Schmitz, M. Weigel, and M. Mühlhäuser. VRSketchPen: Unconstrained Haptic Assistance for Sketching in Virtual 3D Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '20)*, New York, New York, USA, 2020. ACM Press. doi: 10.1145/3385956.3418953.
- [99] C. Zöllner. *Entwurf und Gestaltung eines Hybriden Werkzeugs zum Skizzieren im Dreidimensionalen Raum*. Diploma thesis, Hochschule für Technik und Wirtschaft Dresden, University of Applied Sciences (FH), Dresden, 2007.
- [100] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Greal. Systematic Literature Review of Hand Gestures Used in Human Computer Interaction Interfaces. *International Journal of Human Computer Studies*, 129(March):74–94, 2019. ISSN 10959300. doi: 10.1016/j.ijhcs.2019.03.011.
- [101] S. Schkolne, M. Pruett, and P. Schröder. Surface drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*, pages 261–268, New York, New York, USA, 2001. ACM Press. ISBN 1581133278. doi: 10.1145/365024.365114. URL <http://portal.acm.org/citation.cfm?id=365114><http://portal.acm.org/citation.cfm?doid=365024.365114>.
- [102] F. Quek, D. McNeill, R. Bryll, S. Duncan, X. F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari. Multimodal Human Discourse: Gesture and Speech. *ACM Transactions on Computer Human Interactions*, 9(3):171–193, September 2002. ISSN 1073-0516. doi: 10.1145/568513.568514. URL <https://doi.org/10.1145/568513.568514>.
- [103] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, page

- 1063–1072, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324731. doi: 10.1145/2556288.2557130. URL <https://doi.org/10.1145/2556288.2557130>.
- [104] M. Kavakli and D. Jayarathna. Virtual Hand: An Interface For Interactive Sketching In Virtual Reality. *Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet*, 1(May):613–618, 2005. doi: 10.1109/cimca.2005.1631331.
- [105] J. Huang and R. Rai. Conceptual Three-dimensional Modeling Using Intuitive Gesture-based Midair Three-dimensional Sketching Technique. *Journal of Computing and Information Science in Engineering*, 18(4):1–13, 2018. ISSN 15309827. doi: 10.1115/1.4040982.
- [106] G. C. Burdea. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, Inc., USA, 1996. ISBN 0471021415.
- [107] R. R. Mohanty, R. M. Castillo, E. D. Ragan, and V. R. Krishnamurthy. Investigating Force-Feedback in Mid-Air Sketching of Multi-Planar Three-Dimensional Curve-Soups. *Journal of Computing and Information Science in Engineering*, 20(1), 2020. ISSN 1530-9827. doi: 10.1115/1.4045142.
- [108] A. Gatouillat, A. Dumortier, S. Perera, Y. Badr, C. Gehin, and E. Sejdić. Analysis Of The Pen Pressure And Grip Force Signal During Basic Drawing Tasks: The Timing And Speed Changes Impact Drawing Characteristics. *Computers in Biology and Medicine*, 87(May):124–131, 2017. ISSN 18790534. doi: 10.1016/j.combiomed.2017.05.020.
- [109] J. H. Israel. *Hybride Interaktionstechniken des Immersiven Skizzierens in frühen Phasen der Produktentwicklung*. PhD thesis, Technische Universität Berlin, 2010. URL <http://opus.kobv.de/tuberlin/volltexte/2010/2679/>.
- [110] B. Jackson and D. F. Keefe. Sketching Over Props: Understanding and Interpreting 3D Sketch Input Relative to Rapid Prototype Props. In *IUI 2011 Sketch Recognition Workshop*, 2011. URL <http://ivlab.cs.umn.edu/pdf/Jackson-2011-SketchingOverProps.pdf>.
- [111] K. C. Kwan and H. Fu. Mobi3dsketch: 3D Sketching in Mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2019.
- [112] C. Boddien, J. Heitmann, F. Hermuth, D. Lokiec, C. Tan, L. Wölbeling, T. Jung, and J. H. Israel. SketchTab3d: A Hybrid Sketch Library Using Tablets and Immersive 3D Environments. In *Proceedings of the 2017 ACM Symposium on Document Engineering, DocEng '17*, page 101–104, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346894. doi: 10.1145/3103010.3121029. URL <https://doi.org/10.1145/3103010.3121029>.

- [113] G. Fitzmaurice, J. Matejka, I. Mordatch, A. Khan, and G. Kurtenbach. Safe 3D Navigation. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games, I3D '08*, page 7–15, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939838. doi: 10.1145/1342250.1342252. URL <https://doi.org/10.1145/1342250.1342252>.
- [114] J. Rekimoto. Traxion: A Tactile Interaction Device with Virtual Force Sensation. In *ACM SIGGRAPH 2014 Emerging Technologies, SIGGRAPH '14*, pages 25:1–25:1, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2961-3. doi: 10.1145/2614066.2614079. URL <http://doi.acm.org/10.1145/2614066.2614079>.
- [115] S. Günther, M. Makhija, F. Müller, D. Schön, M. Mühlhäuser, and M. Funk. PneumAct: Pneumatic Kinesthetic Actuation of Body Joints in Virtual Reality Environments. In *Proceedings of the 2019 on Designing Interactive Systems Conference, DIS '19*, pages 227–240, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5850-7. doi: 10.1145/3322276.3322302. URL <http://doi.acm.org/10.1145/3322276.3322302>.
- [116] P. Lopes, A. Ion, and P. Baudisch. Impacto: Simulating Physical Impact by Combining Tactile Stimulation with Electrical Muscle Stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15*, pages 11–19, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3779-3. doi: 10.1145/2807442.2807443. URL <http://doi.acm.org/10.1145/2807442.2807443>.
- [117] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson. Dexmo: An Inexpensive and Lightweight Mechanical Exoskeleton for Motion Capture and Force Feedback in VR. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16*, pages 1991–1995, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858487. URL <http://doi.acm.org/10.1145/2858036.2858487>.
- [118] by 3D Systems. Touch. <https://www.3dsystems.com/haptics-devices/touch>, 2020.
- [119] by 3D Systems. Phantom. <https://www.3dsystems.com/haptics-devices/3d-systems-phantom-premium>, 2020.