Mobi3DSketch: 3D Sketching in Mobile AR

Kin Chung Kwan
School of Creative Media, City University of Hong Kong
kckwan@cityu.edu.hk

Hongbo Fu
School of Creative Media, City University of Hong Kong
hongbofu@cityu.edu.hk

(a) 3D sketching in mobile AR (b) “Phoenix” (34.2 mins) (c) “Spiderman” (36.3 mins)

Figure 1: *Mobi3DSketch* is designed for users with reasonably good drawing skills to create 3D concept designs in the context of real-world environments using a single AR-enabled mobile device (a). (b) and (c) are 3D concepts created with *Mobi3DSketch* in situ. The small figures show the results from other viewing angles.

ABSTRACT

Mid-air 3D sketching has been mainly explored in Virtual Reality (VR) and typically requires special hardware for motion capture and immersive, stereoscopic displays. The recently developed motion tracking algorithms allow real-time tracking of mobile devices, and have enabled a few mobile applications for 3D sketching in Augmented Reality (AR). However, they are more suitable for making simple drawings only, since they do not consider special challenges with mobile AR 3D sketching, including the lack of stereo display, narrow field of view, and the coupling of 2D input, 3D input and display. To address these issues, we present *Mobi3DSketch*, which integrates multiple sources of inputs with tools, mainly different versions of 3D snapping and planar/curves surface proxies. Our multimodal interface supports both absolute and relative drawing, allowing easy creation of 3D concept designs in situ. The effectiveness and expressiveness of *Mobi3DSketch* are demonstrated via a pilot study.

CHI 2019, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300406

CCS CONCEPTS

• **Human-centered computing** → *Mixed / augmented reality*; **Systems and tools for interaction design**; **Ubiquitous and mobile computing systems and tools**.

KEYWORDS

3D sketching; augmented reality; mid-air drawing; mobile interaction; relative drawing; 3D concept design

ACM Reference Format:
Kin Chung Kwan and Hongbo Fu. 2019. Mobi3DSketch: 3D Sketching in Mobile AR. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3290605.3300406

1 INTRODUCTION

Mid-air 3D sketching allows artists to draw 3D virtual curves directly in the air. This has been mainly explored in Virtual Reality (VR) and requires the use of immersive, stereoscopic display systems and outside-in motion capture systems for tracking 3D styluses [10, 20]. In recent years, the consumer-level display and motion tracking hardware (e.g., HTC Vive) for VR make 3D sketching (e.g., by using Google Tilt Brush) more accessible by artists, though special equipment and complicated setup are still needed. Meanwhile the recently developed motion tracking algorithms such as Concurrent Odometry and Mapping (COM) [30] and Visual Inertial Odometry (VIO) [25] allow real-time inside-out 3D motion tracking of mobile devices by using their cameras and IMUs alone. Such algorithms have been integrated into the modern Augmented Reality (AR) platforms such as ARKit by Apple and ARCore by Google, making 3D sketching in mobile AR more feasible and easily accessible.

Although several mobile applications for 3D sketching (e.g., *AR+Brush* [28], *Paint Space AR* [3], *Just a Line* [14]) have been developed, they have been used to create simple doodling for fun and are difficult to make 3D concept designs. This is largely because they are designed by mainly following existing VR 3D sketching systems, without attempting to address the special problems related to 3D sketching in mobile AR, including the lack of stereo display, narrow field of view, and the coupling of 2D input, 3D input and mobile display. Such challenges easily cause imprecise 3D sketches and poor connectivity between 3D curves. In addition, none of them explicitly considers how to interact with real environments, though such a feature is critical for practical AR applications.

We present *Mobi3DSketch*, a novel 3D sketching system for designers to create 3D concept designs in a real-world context using a single AR-enabled mobile device. To tackle the above challenges, we adapt the commonly used tools, namely 3D snapping and surface proxies, originally designed for 3D sketching with 2D inputs. Our system supports explicit snapping and implicit snapping with snapping suggestions, snapping curves/surfaces, and one/two-endpoint snapping so that users may easily create sketches with well connected strokes. We exploit a mobile device's 3D pose to support quick anchoring of planar proxies and easy creation of extruded surfaces in space. Such surface proxies either manually anchored or obtained from environment understanding allow easy attachment of 3D sketches to real-world environments. Although most of these techniques have already been individually studied in many applications [9, 11, 19, 31, 34], we carefully unify them into a powerful workflow to develop a mobile AR 3D sketching system. Our unified features for snapping and surface proxies not only allow artists to sketch with greater precision but also to seamlessly switch between absolute drawing and relative drawing. The latter is very important for sketching large-scale objects (Figure 1) through a narrow field of view, and has seldom been explored in existing VR/AR-based 3D sketching systems, which often use the absolute positions of a continuously tracked 3D stylus for defining 3D curves.

Our multimodal interface incorporates multiple sources of inputs from a mobile device to seamlessly integrate mid-air 3D sketching with surface-based 2D sketching. Unlike traditional mid-air 3D sketching systems in VR/AR, which have been limited to a room with special hardware setup, *Mobi3DSketch* works well both indoors and outdoors. In Figure 1 we show our representative sketching results, which, with this level of complexity, have never been demonstrated by existing mobile AR 3D sketching applications. We quantitatively evaluate the effectiveness of individual components of *Mobi3DSketch*, and demonstrate the expressiveness of our system by showing various 3D in-situ designs created by multiple users.

Our contributions are summarized as follows: (1) challenge analysis for mobile AR 3D sketching, (2) the design of novel interaction, (3) the first working prototype, and (4) pilot studies with compelling results.

2 RELATED WORK

3D Sketching in VR. Mid-air 3D drawing or surface modeling has been explored firstly as desktop applications [34] and then as immersive VR applications [10, 20, 35]. It often requires special equipment such as head-mounted displays (HMD) or CAVE-like systems for immersive, stereoscopic visualization, and 3D styluses (or similar devices) for direct 3D inputs. Both the head of a user and a stylus need to be accurately tracked to align the input and output spaces. Although such setups potentially enable various applications like 3D concept design [35], the experiments by Arora et al. [1] show that precise 3D sketching in mid-air is challenging mainly because of the limited human ability to sketch freely in 3D without a physical supporting surface. This problem might be alleviated by using haptic constraints [12, 21] or reference imagery [18]. The recent development of VR HMD and motion-tracked controller hardware (e.g., HTC Vive, Oculus Rift) and software (e.g., Tilt Brush by Google, Gravity Sketch) has made mid-air 3D sketching more accessible by users. However, due to the cost of the special VR hardware and its rather complicated setup, VR-based 3D sketching is still limited to a small group of users. In contrast, our goal is to make 3D sketching more accessible and widespread, and our focus is on a mobile AR 3D sketching interface for 3D concept design in situ.

3D Sketching in AR. Compared to VR, AR allows users to author 3D sketches that are more directly linked to real-world objects or environments, and thus requires reasonably accurate tracking of users and/or objects in a physical space [29]. For example, Yee et al. [42] propose a video see-through AR system for mid-air 3D sketching using an HMD with cameras and a drawing pen, both of which are tracked by using a commercial MoCap system. A similar system setup is adopted by Tano et al. [38] for authoring life-sized 3D sketches. Observing imprecise mid-air sketching [1], Arora et al. [2] present *SymbiosisSketch*, which combines 3D mid-air sketching using a motion-tracked stylus and 2D surface sketching on a tablet, and displays sketching results through an AR HMD (Microsoft HoloLens). Due to their essential use of outside-in MoCap systems, the above AR systems are more like laboratory-based implementations. Even if these systems adopt inside-out motion tracking techniques, they still cannot be easily adapted to our problem of 3D sketching using a single mobile device, due to the challenges discussed in Section 3. AR markers provide an alternative way to register the real and the virtual worlds. For example, the *Napkin Sketch* system [39] allows users to draw 3D sketches in a small working volume on top of AR makers using a projective 3D sketching approach, which is analogous to our system. However, they are in fact very different. Napkin Sketch is essentially "object-centered", and requires the camera to always look at a sketched object being created. Thus their system is more suitable for creating small-scale sketched objects, rather than large-scale 3D sketches situated in a real-world environment. Our problem demands a "viewer-centered" approach, resulting in different challenges and solutions.

The recent motion tracking techniques such as COM [30] and VIO [25] rely on both the visual information from a mobile device's color camera and inertial measurements from the device's IMU (Inertial Measurement Unit) sensors to robustly estimate the 3D pose (3D position and 3D orientation) of the camera (and thus the device) in real time. Based on these techniques, in 2017 Apple and Google respectively released ARKit (based on VIO) and ARCore (based on COM), modern mobile AR platforms. With ARKit or ARCore, users may use a mobile device as a 3D stylus for mid-air 3D sketching. While multiple mobile applications such as *AR+Brush* [28] and *Paint Space AR* [3] have been developed for experimenting 3D sketching on these mobile AR platforms, all of them are created for fun, instead of more serious tasks like 3D concept design. They mainly use a mobile device as a 3D stylus for mid-air 3D sketching, similar to the existing VR/AR 3D sketching systems, and have not attempted to address the special challenges of mobile AR-based 3D sketching.

3D Sketching with 2D Inputs. 2D sketching is still one of the most efficient ways to express ideas, since the traditional input devices like computer mice, graphic tablets, touchscreens support 2D inputs only. There is a large body of research that focuses on how to lift 2D sketches into 3D. This problem is challenging due to its ill-posed nature, since theoretically there are an infinite number of 3D interpretations for a given 2D sketch drawn from a specific viewpoint. Different kinds of regularity constraints like planarity, orthogonality, symmetry have been proposed to solve for an optimal 3D interpretation of a carefully drawn, complete 2D sketch [27, 40, 44]. Alternatively, 2D strokes can be lifted into 3D one by one interactively, for example by making use of planar/curved 3D surfaces for stroke projection [4, 7, 11, 15, 16, 19, 22, 26, 32, 41], scaffolds and geometric priors [23, 36, 43]. Similar to *SymbiosisSketch* [2], Kim et al. [22] combine 2D inputs (pen drawing) and 3D inputs (unconstrained 3D hand motions for air scaffolding) for 3D sketching, but as a desktop application. Some of the above techniques have been employed by AR-based sketching techniques [5, 39], which, however, focus on 3D sketching on top of AR markers using 2D input devices. Our work moves towards in-situ 3D sketching of larger-scale objects and explores how to integrate multiple sources of *2D and 3D inputs*, with some of the existing tools, mainly surface canvases [4, 11] and various versions of snapping [6, 36], for a multimodal 3D sketching interface in mobile AR.

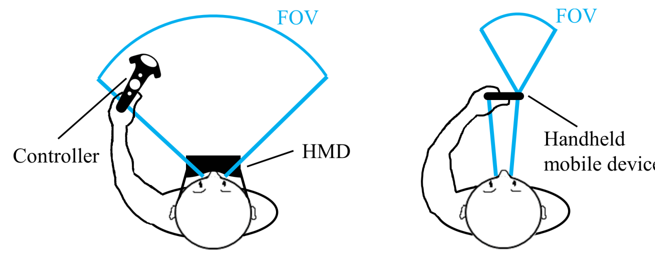


Figure 2: Small-sized mobile screens result in a much narrower field of view (FOV) than VR HMDs. 2D/3D input and display are coupled in our case.

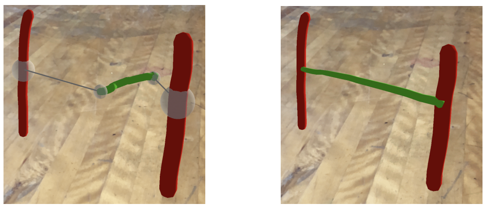
Multimodal Mobile Interaction. Handheld mobile devices such as smartphones and tablets are equipped with more and more sensoring capabilities for example via multi-touch screens, IMU sensors, depth cameras etc. It has been demonstrated that integrating multiple sources of inputs can benefit various 3D modeling applications [8]. For example, *MobiSweep* [33] combines direct orientation control (based on the orientation of a mobile device) with indirect position control (via multi-touch gestures) for 3D sweep surface modeling. *Window-Shaping* [17] uses multi-touch interaction for sketch-based 3D modeling on and around physical objects sensed by an integrated depth sensor. Besides handheld devices, some other systems require the use of an additional motion tracking infrastructure [24] and/or immersive display [12]. Our work uses a mobile device alone for a different application of 3D sketching.

3 CHALLENGES

We aim for a mobile AR 3D sketching interface to make conceptual designers easily create 3D sketches in situ and examine them in different view angles. This is impossible with traditional 2D or 3D sketching systems unless the real-world context is modeled digitally (a rather challenging problem on its own) in advance. Simply using an AR-enabled mobile device as a 3D pen for mid-air drawing, as done similarly in the existing AR/VR 3D sketching systems and mobile applications, does not allow easy creation of 3D concept design in situ.

In fact, we observe at least the following challenges for mobile AR 3D sketching. 1) It is difficult to precisely control the 3D position and orientation of a device in mid-air without any physical supporting surface. 2) The displays of most

Figure 3: (Left) Lack of context due to narrow FOV. (Right) Users can step back to perform relative drawing while seeing the whole context.



(a) Before snapping (b) After snapping

Figure 4: Snapping suggestions (in white) to connect the endpoints of a new stroke (in green) to the existing ones (in red).

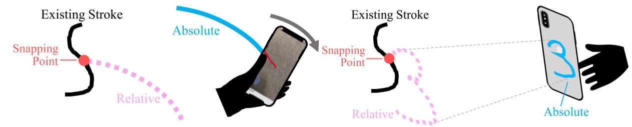
of the mobile devices are not stereoscopic. This makes users difficult to perceive the depth of 3D strokes on the screen, resulting in drawings with wrong depth. 3) Small-sized mobile screens result in an extremely narrower field of view (FOV) compared to immersive VR displays (Figure 2), and a small operation area for multi-touch interaction. This would easily make users draw poorly without seeing the whole context, as illustrated in Figure 3. 4) The output and 2D/3D input of mobile device are coupled. This coupling makes users difficult to check whether the depth of a stroke being currently drawn is reasonable or not by examining it from another viewpoint. Steadily holding the device in mid-air easily leads to fatigue or errors otherwise due to device shaking.

4 DESIGN GOALS

Our interface is designed to alleviate the problems associated with the above challenges. We have the following design goals.

Sketching with Greater Precision. Users are hard to determine the depth of a stroke being drawn from the current viewpoint due to the lack of a stereo mobile display and cannot check its depth from another viewpoint while drawing due to the coupling of the input and the output. This often leads to 3D curves that appear connected in the screen space but separate in 3D (Figure 4 (a)). Moreover, due to the small-sized multi-touch screen and the well-known fat finger problem [37], the strokes may not be connected accurately even in the screen space. One of our goals is thus to adapt some useful tools including surface proxies and 3D snapping to suppress connection errors caused by these problems.

Support of Relative Drawing. Most existing mid-air 3D sketching systems have focused on the use of a tracked stylus for *absolute* 3D positioning of a virtual 3D brush. However, to tackle the narrow FOV challenge and create large-sized 3D sketches in mobile AR, users typically have to step away from



(a) Mid-air 3D drawing (b) Surface-based 2D drawing

Figure 5: With and without using a snapping point, users can easily switch between relative drawing (pink dotted lines) and absolute drawing (blue line).

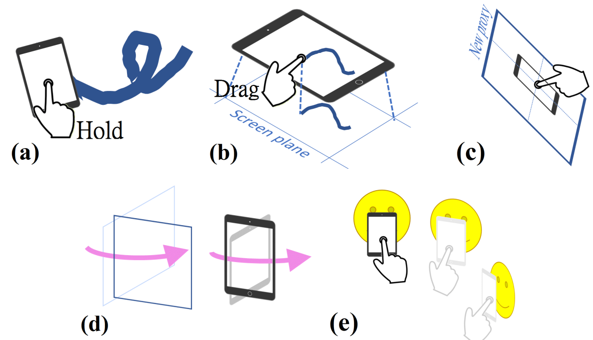


Figure 6: Representative operations in our multimodal interface for mobile 3D sketching. (a) Mid-air 3D drawing, (b) surface-based 2D drawing, (c) proxy creation, (d) proxy rotation, and (e) cloning.

virtual objects (e.g., a tree) until they see a global context for drawing. The sketches then might be beyond the reach of arm. This motivates us to support the mode of relative drawing (Figure 5) to allow users to draw sketches distantly. This requires us to use the 3D position from motion tracking relatively, with respect to one or multiple points in an existing sketch.

Interaction with Real Environments. One of the main advantages with mobile AR 3D sketching is to create 3D designs with respect to real objects or scenes. Both ARKit and ARcore have limited ability to understand the surrounding environment by detecting planar structures in a scene. We aim to use such automatically detected planes or allow interactive creation of similar surface proxies to achieve 3D sketches in situ (Figure 19).

5 SYSTEM OVERVIEW

Our mobile device together with the modern AR platform (ARKit in our case) provides multiple sources of input, from multi-touch screen, motion tracking, IMU sensors, etc. Since these input sources have their own characteristics, we integrate them and design a multimodal interface to fulfill our design goals.

Figure 6 shows representative operations in our multimodal interaction for 3D sketching. More details will be given in Section 6. To create 3D sketches of reasonably good quality, we adapt two useful tools including surface proxies and 3D snapping, which are originally designed for 3D sketching with 2D inputs but seldom used for mobile 3D sketching.

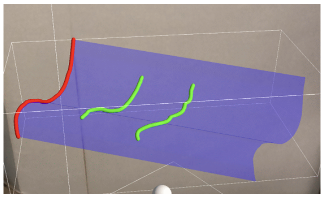


Figure 7: Curved proxy.

Figure 8: Snapping surface.

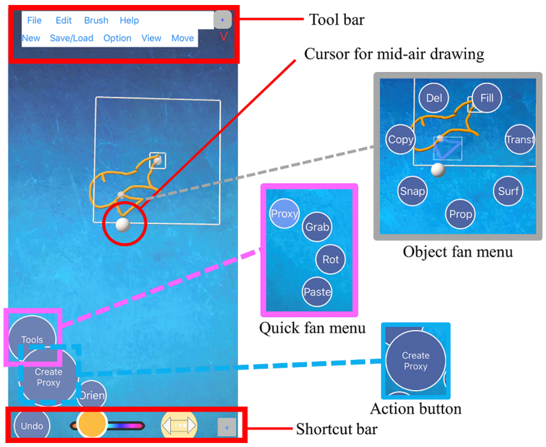


Figure 9: The user interface of *Mobi3DSketch*. The functions of the action button change with the selected tools.

Planar/curved surface proxies can help users draw 3D strokes more precisely by constraining them onto the proxy surfaces (Figure 7). Such surface proxies also help suppress the depth errors caused by device shaking while drawing. Relative drawing or large-scale sketching can be easily supported by drawing on the proxies with perspective projection.

While there is no easy way to guide precise drawing of individual unconstrained 3D freeform curves, we use *endpoint snapping* to achieve well connected strokes. The snapping feature has already been adopted by *Paint Space AR* [3], which, however, only supports the snapping of the starting point of a 3D stroke to any point of the existing sketch. We provide a solution to another commonly used snapping scenario, where the endpoints of a stroke both need to be snapped to the existing sketch (Figure 4). This allows users to connect a new stroke to two existing 3D strokes with ease.

To achieve relative drawing, users may explicitly indicate their relative drawing intention by first specifying a snapping point on an existing stroke or surface proxy before drawing a new stroke (Figure 5). Fortunately, no explicit mode switching between absolute and relative drawing is needed, since both of the modes can be achieved by using the same mechanisms of 3D snapping and surface proxies.

Figure 9 shows the main interface of our system. We place a shortcut bar at the bottom of the screen, and organize commonly used operations into the quick fan menu placed at the corner of the device. If a tool (e.g., "Proxy") is selected in the fan menu, it appears as an Action button (e.g., "Create Proxy") so that users can easily repeat the same operation for multiple times. This design is especially useful for the operations that require frequent usage (e.g., cloning an object for multiple times) or fine tuning (e.g., translation). If users long-press an already-selected virtual object (i.e,. snapping point, stroke, surface proxy) on the 2D screen, an object fan menu pops up and allows users to perform specific operations, such as delete or fill color, to the selected object.

Operation	Input	Description
Selection	touch	Single-finger tap for selecting a stroke or snapping point; two-finger tap for selecting a proxy. A proxy can also be selected by single-finger tapping its center. Tap on any empty space for deselection.
Translation	MoT, touch	A selected object moves with the device in space when the "Grab" action button being pressed. An optional constraint for vertical translation is also provided.
Rotate	IMU, touch	A selected object rotates according to the orientation of the device (Figure 6 (d)) when the "Rotate" action button being pressed.
Scaling	touch	Two-finger pinch gesture for scaling up/down a selected object.
Zoom-in/out	IMU, touch	Two-finger pinch gesture for moving a selected object along the device's viewing direction when the "Grab" action button being pressed.
Cloning	MoT, touch	Copy a selected object, and place it in front of the device (Figure 6 (e)) when the "Paste" action button is tapped.
Deletion	touch	Long-press a selected stroke, proxy, or snapping point and then choose "Del" in the pop-up fan menu.

Table 1: A list of supporting operations. Objects here mean individual strokes, groups of strokes, or proxies. "MoT" stands for motion tracking.

A list of supporting operations using multimodal interaction can be found in Table 1. A user is suggested to use his/her dominant hand to perform drawing or touch gestures on the screen, and the non-dominant hand for holding the device and interacting with the shortcut bar and the quick fan menu. This design for bimanual interaction allows quick or temporary mode switching during sketching.

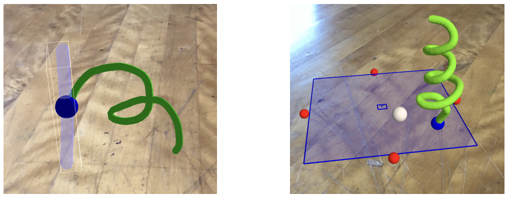
6 USER INTERFACE

We first introduce three basic components of our system, and then discuss their use for 3D sketching.

6.1 Basic Components

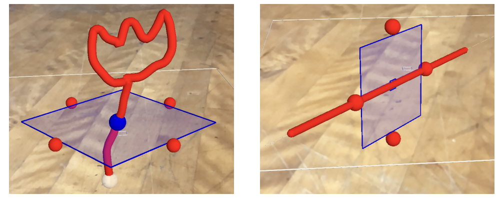
3D Strokes. Strokes in our system are rendered as generalized cylinders (Figure 10). To reduce tracking noise and motor control errors, we smooth strokes using the same method presented in [2]. A stroke can be selected or deselected (see Table 1). Selected strokes are shown semi-transparently.

Snapping Points. Snapping points are rendered as small spheres, with the active one shown in solid blue and others



(a) From an existing stroke (b) From an existing proxy

Figure 10: Relative drawing from explicitly specified snapping points (blue points) on a selected stroke or proxy.



(a) Attach to a point (b) Attach to a line

Figure 11: Create proxies attached to an existing point or line.

in semi-transparent white. To specify a snapping point, the user taps on a specific position of a selected stroke or proxy (Figure 10). If two strokes are linked with the same snapping point, any transformation operation on one stroke will be applied to the other so that their connectivity remains the same after transformation. The deletion of the snapping point breaks the link between the two strokes.

Surface Proxies. Our system supports both planar and curved surface proxies. To create a planar proxy, the user selects the "Proxy" tool in the quick fan menu and taps the "Create Proxy" action button. One planar proxy for each click of the action button is instantly created according to the current 3D pose of the device (Figure 6 (c)), without a sequence of plane definition and adjustment steps [11, 39]. In other words, the newly created proxy will lie on the plane of the device. This seamless creation process is very useful for the user to create multiple proxies aligned with different planar surfaces of real-world objects. Alternatively, the user may first select an existing stroke or a snapping point on an existing stroke or proxy, and then create a planar proxy passing through the two endpoints of the selected stroke or the selected snapping point (Figure 11). Since unconstrained 3D orientation control is often imprecise, we provide *orientation snapping* to existing proxies, and the horizontal and vertical orientations. The displayed size of a planar proxy can be adjusted by long-pressing and dragging its four control handles (red spheres in Figure 11).

Our curved surface proxy is an extruded surface (Figure 7). To enter the surface creation mode, the user long-presses a selected 3D stroke, and select the "Surf" button in the object fan menu. The user then extrudes the selected 3D stroke as a profile along a straight line or freeform curve by mid-air drawing. Planar/curved surface proxies are rendered as semi-transparent 3D surfaces. To make them become essential parts of a design, the user may fill them with a solid color through the "Fill" item in the object fan menu.

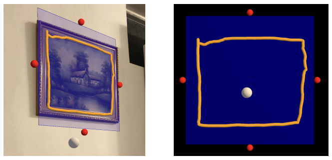
6.2 Main Interaction

Sketching. The user may perform either mid-air 3D sketching (Figure 6 (a)) or surface-based 2D sketching (Figure 6 (b)). For mid-air sketching, the user utilizes the device as a 3D pen (cursor for drawing placed at the screen center (Figure 9)) and moves the device in space while long pressing anywhere on the screen. Surface-based sketching is achieved by a single finger moving on the touch screen. During sketching, the action button becomes a "Freeze" button. The user may press this button to pause (or freeze) the current drawing process while changing the viewpoint, and release the button to resume the drawing process. To seamlessly support both mid-air sketching and surface-based sketching, our system automatically determines the sketching mode by checking if a device or a finger touching on the screen moves first, by thresholding the changes of touch position and device position.

Relative Drawing on Snapping Point. For a given selected snapping point, in the mid-air 3D sketching mode, all the new strokes will be drawn starting from the selected snapping point (Figure 10). This is achieved by moving the starting point of a new stroke to the snapping point. For surface-based, to prevent operations by mistake, the user is required to draw a new stroke starting at or near to the selected snapping point in the 2D screen space, determined by ray-line intersection.

Relative Drawing on Surface Proxy. The user directly draws strokes on a selected proxy using surface-based sketching from any distance and any orientation. The new strokes will be back-projected onto the (planar or non-planar) proxy through perspective projection. One interesting feature of a planar proxy is that the user can draw outside the proxy, since there is essentially an infinitely large plane behind each planar proxy. This feature is particularly useful for creating large-scale sketches. In addition, to avoid fatigue during surface-based sketching while holding the device and pointing to the selected proxy, the user may enter the "2D" mode by selecting the "2D" item in the object fan menu after long-pressing a selected surface proxy, and then draw on the 2D parameter domain of the surface proxy (Figure 12) while holding the device more comfortably. A similar interface has been explored in *SymbiosisSketch*. This "2D" mode can also used to bring a remote proxy closer to the user and make it parallel to the image plane, thus improving the precision for drawing distant sketches or drawing on proxies that are not parallel to the screen.

In-context Sketching. To create 3D sketches attached to a real-world environment, the user may use environment



(a) Normal mode (b) "2D" mode

Figure 12: Draw in the 2D parameter domain (b) to avoid fatigue during drawing while pointing to a proxy.

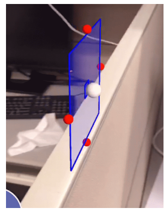


Figure 13: Create a planar proxy that aligns with a real plane.

planes automatically detected by environment understanding as planar surface proxies (Figure 12 (a)). When such environment planes do not exist or cannot be detected, the user may move the device close to a real object, and create an anchor (i.e., 3D stroke or proxy) for the attachment of other strokes. However, due to the inside-out motion tracking technique, the camera of the device cannot be moved too closely to the real object. Otherwise the motion tracking algorithm would fail. To address this issue, the user may create a planar proxy that passes through desired 3D positions at the place where motion tracking works well (Figure 13), and then specify an anchor on this proxy. Note that this issue does not exist with *SymbiosisSketch* [2] due to its adoption of outside-in MoCap.

Snapping Recommendation. If the user does not specify any snapping point explicitly, our system provides snapping suggestions if appropriate. More specifically, if an endpoint (either starting or ending) of a new 3D stroke and any point of the existing sketch are very close to each other in the screen space, there is a high possibility that the user intends to snap the two strokes. Our system provides a snapping suggestion, and the user may accept the suggestion by clicking the "Snap" action button or deny it by tapping on an empty screen space. It is also possible to use snapping suggestions for the both endpoints of a new stroke (Figure 4). The depth of intermediate points of the stroke is linearly interpolated from the depth of the endpoints. The user may adjust snapping suggestions by dragging the suggested snapping points to change their positions or tapping to remove them. Such refinements of snapping suggestions often need to be done in a viewpoint different from the original viewpoint for drawing the new stroke.

Snapping Curve & Snapping Surface. Beside individual snapping points, we also introduce snapping curves and snapping surfaces, which can be considered as regular strokes or surface proxies containing many pre-defined snapping points on them. When a new stroke is drawn near a snapping curve or surface, one of its endpoints will be automatically snapped to the nearest point of the snapping curve or surface. This feature is particularly useful when a lot of strokes need to be attached to an existing stroke or surface proxy, e.g., to add hair curves to a head surface (Figure 8). We use the space-based distance measurement and screen-based distance measurement for mid-air 3D sketching and surface-based 2D sketching, respectively. To change a selected stroke (proxy) to a snapping stroke (surface), the user chooses the "Snap" item from the object fan menu.

Other Tools. One additional interesting tool in our system is a symmetry proxy that can regularly and accurately duplicate the currently drawn stroke to different positions or orientations. Both reflective and rotational symmetry proxies are provided. We also provide 3D transformation widgets (similar to those in CAD software) for transforming the existing strokes and proxies more accurately. They allow users to create a detailed object by first drawing it in a relatively bigger scale and then scaling it down. Similar to most painting systems, our system also provides basic but important tools including color picking, undo & redo, primitive drawing, and save & load files, etc.

7 USER EVALUATION

We have implemented our prototype based on Apple ARKit. For the evaluation below, we mainly used iPhone with fingers for multi-touch interaction. We also tested our prototype on iPads but they are too heavy for mid-air 3D sketching.

7.1 Evaluation

Since absolute drawing in our system is almost the same as that in existing mobile apps for 3D sketching, we focused on the evaluation of our new features (relative drawing with snapping and proxy). We hypothesized that relative drawing is more preferable in some common tasks. We included five different drawing methods in this pilot study, including 3D mid-air absolute drawing (**MA**), 3D mid-air relative drawing (**MR**), surface relative drawing (**SR**) [3], surface drawing with two-endpoint snapping (**TS**), and surface drawing with planar proxy (**PR**).

We invited 10 university students (age 21-30, 3 female) to test our system by performing a set of fixed tasks (without requiring drawing skills). Before the evaluation, we provided a 5-minute tutorial to let the subjects get familiar with the goal and operations of the five methods. During the study, subjects were asked to complete four different tasks for five times with all the five methods. Latin squares were used for the order of the methods amongst the subjects to minimize bias from learning.

Tasks. The four tasks were: (1) "Circle" task involved the drawing of a 2D perfect circle in the mid-air. This task was to evaluate whether the subjects could draw a desired shape well using different methods. (2) "Link" task involved the drawing of a straight line to connect two virtual objects in the mid-air. The motivation of this task was to evaluate whether the subjects could connect existing objects in mid-air well. (3) "Ladder" task involved the drawing of a 4-step ladder.

The goal was to test subjects with drawing objects involving multiple strokes. (4) "Tall-line" task involved the drawing of a very tall vertical line from a given starting point to see how big scale the user could manage to draw using different methods.

Procedure. All these tasks were performed on an iPhone X with iOS 11.4 in a normal lab environment with normal lighting condition. Our goal in this study was to evaluate the drawing performance using our provided methods. Thus we provided all the required snapping points and proxies in the test if needed (e.g., one snapping point is needed for relative drawing), except the "ladder" task. The subjects were required to create their own snapping points for the horizontal parts of a ladder. At the end of the study, each subject was asked to provide feedbacks.

Results and Feedbacks. Figure 17 shows the projected results of all the subjects in the "circle" and "link" tasks, and Figure 18 the example results of 2 subjects in the "ladder" task for visual comparison. It can be easily seen that snapping and planar proxies play important roles in creating better results. It is very difficult to achieve acceptable results using **MA**. Similar problems exist with the existing mobile AR 3D sketching applications.

In the study the subjects provided lots of feedbacks. Overall all of them felt that relative drawing was useful, and snapping and proxy were helpful for 3D sketching in Mobile AR. As expected, all the subjects complained that they could not see the whole context for absolute drawing and no depth perception on the screen. Several subjects missed the target object in the "link" case and drew random curves instead of a straight line (Figure 17 (bottom)). All the subjects reported that surface-based relative drawing was more preferable than mid-air drawing for these common tasks, since they could see the whole content on the screen. In particular, S2 comments:*As we cannot see the objects outside the small screen, we need to have a very strong sense of 3D space to draw strokes well using absolute drawing.* It is also evidenced by the quantitative measurement in our study. A paired t-test shows surface-based drawing (**SR**, **TS**, **PR**) was significantly better than mid-air drawing (**MA**, **MR**) in terms of both time and error ($p < 0.001$) for all tasks. However, although surface-based drawing is better in these four common tasks, **MA** is still powerful, especially when using the physical environment as the reference. Our system thus supports both absolute and relative drawing, since we believe they have their own strengths in different cases.

Our **TS** also helps improve the quality of stroke connection (Figure 16), even for strokes with different depth. S10 commented: *Connection of strokes is an important aspect in sketching. Two-endpoint snapping is very useful to connect strokes with different depth.* Our study shows that the offset distance between strokes [1] with **TS** is significantly smaller than each of the other methods ($p < 0.001$). Compared to all the other methods, **PR** significantly ($p < 0.001$) reduced the depth errors caused by device shaking, as confirmed by S5: *Proxy is good as it allows us to draw the sketch with same depth on the screen from any distance and any orientation.* In addition, it allowed the subjects to draw large-sized sketches more than 1,000m tall using **PR**, compared to the other methods with the average heights: 110.8cm (**MA**), 164.5cm (**MR**), and 208.1cm (**SR**).

Users also reported that there is no performance difference for drawing objects with varying depth using absolute or relative mid-air drawing. Although they are able to see the whole context on the screen using relative drawing, the size difference caused by perspective projection distorts their expectation, especially for objects with varying depth. They suggested that they could draw better sketches using mid-air drawing if they do not look at the screen. Thus mid-air relative drawing is currently only useful for drawing distant 3D objects in a large-scale scene. It does not provide any benefit on the drawing quality or speed compared to mid-air absolute drawing.

7.2 Design Results

We recruited three student helpers with reasonably good drawing skills to test the expressiveness of our tool. All of them did not have any previous 3D sketching experience. They were given a 1-hour tutorial on our system and were also introduced to *Paint Space AR* [3], which is probably the best available mobile application for 3D sketching. They felt while our tool took much longer time to learn, it was much more powerful than *Paint Space AR*.

Figures 1 and 19 shows representative results created by these student artists. We show various use cases where 3D sketches were created to decorate the real-world objects and environments. Our tool works well for sketches with different scales, from several centimeters to dozens of meters. On average, each design was created about a dozen of minutes. The most complicated examples in Figure 1 took over half an hour for each example.

The preliminary analysis on the sketching process of the participants shows that each main feature played important roles in producing the sketching results. On average 8.43 proxies, 38.57 one-endpoint snapping, 4.57 two-endpoint snapping, 8.79 absolute drawing operations and 75.92 relative drawing operations were used per sketch with on average 66.86 strokes, excluding deleted strokes.

[1] Defined as the 3D Euclidean distance between the starting and ending points for the "circle" task; the 3D Euclidean distance of a starting or ending point to the corresponding virtual object for the "link" task; the average of 3D distance of the endpoints of the horizontal lines to the nearest points in the vertical lines for the "ladder" task.

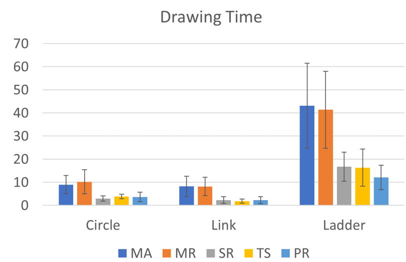


Figure 14: Drawing times.

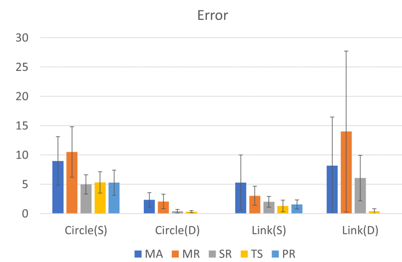


Figure 15: Drawing errors. “S”: shape error, “D”: depth error.

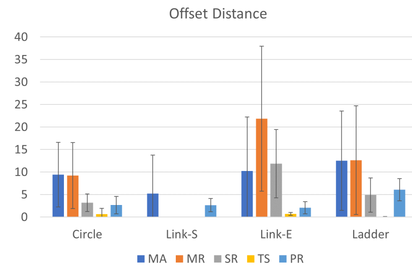


Figure 16: Offset distance. Link-S (E) represents the starting (ending) point.

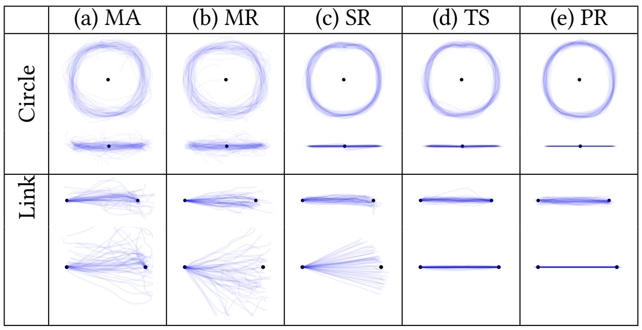


Figure 17: Projected results from different subjects for different methods in the “circle” and “link” tasks. The upper row of each case is the image view, while the bottom row is the top view, showing depth variance. The dots inside represent the circle center and the linking objects.

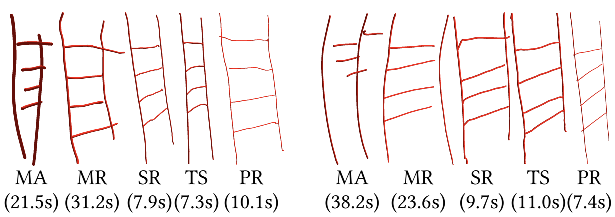


Figure 18: Example results of two selected subjects for the “ladder” task with different methods in our study. The times under the image are their corresponding drawing time.

All of the helpers appreciated the features of our tool. One participant manually created a proxy that aligned with a physical partitioner (Figure 13) and used to grow tree branches in Figure 1 (b), since ARKit failed to detect any plane there. Figure 19 (b) shows the necessity of using relative drawing to attach a hammock to the physical trees. The attached points were too high for any absolute drawing method. Our proxy allow users to draw a reference line near the points and use two-end snapping to connect it with the hammock. The proxy is also useful to draw multiple disconnected strokes on the same surface. For instance, the augmented decoration lines on the real chair in Figure 19 (h) were created on the same proxy. Mid-air absolute drawing is useful to draw lines when using the physical objects as the reference. For instance, one participant used it to draw reference lines to represent the heights of human body parts, in order to draw a 1:1 scale human (Figure 1 (c)). They also found useful to use strokes with large width to simulate 3D effects by Tilt Brush (see Figure 1).

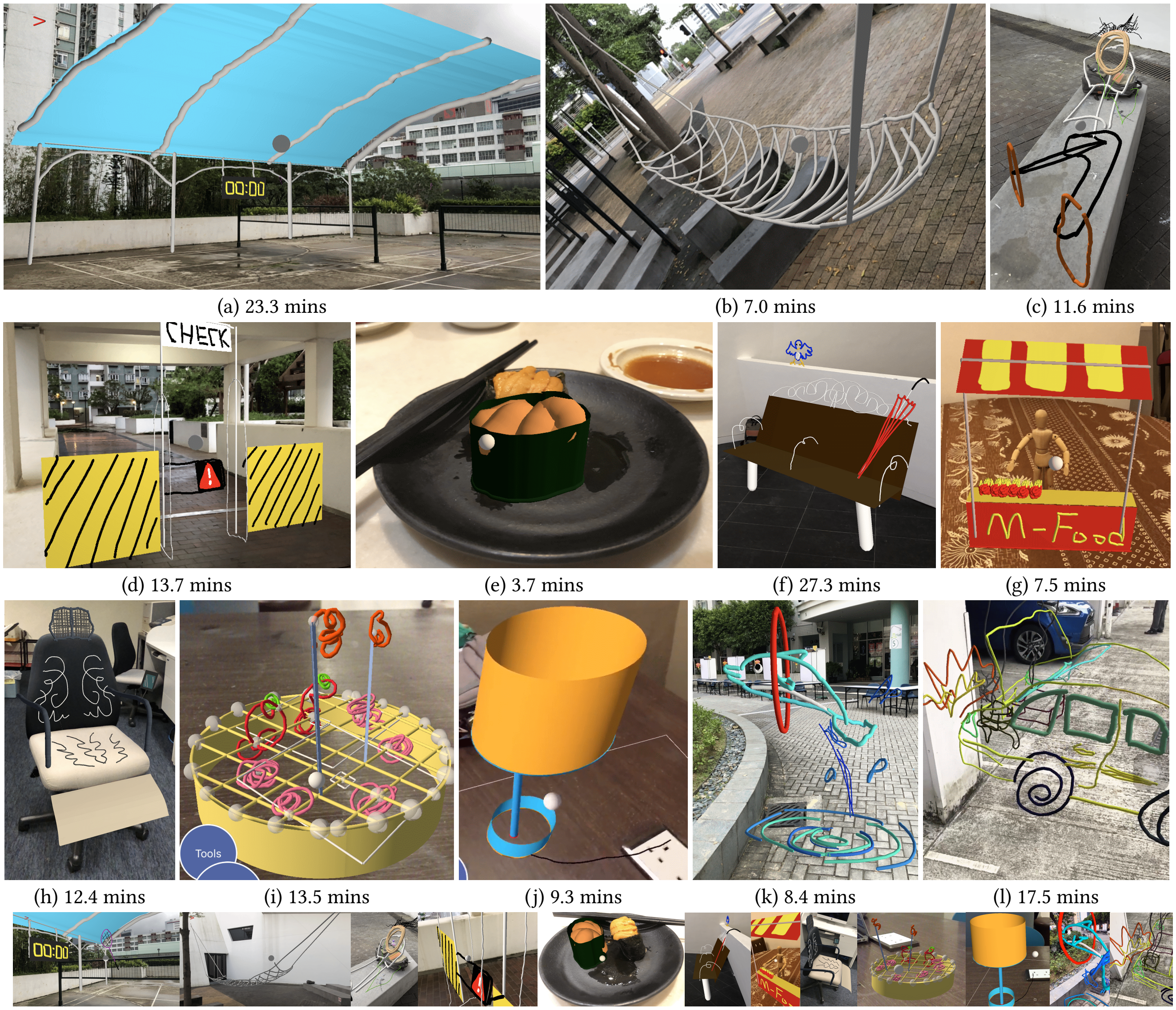
3D sketching in mobile AR is a rather new and largely unexplored problem. We have shown that it is particularly useful for 3D concept design in the context of a real-world environment. Unlike traditional 3D modeling software, our system allows users to quickly create and examine 3D concept sketches that closely interact with the environment, as demonstrated in our results. Most of our results are difficult or much more time-consuming to make with the existing mobile AR-based 3D sketching applications.

Since we use a hand-held device alone for 3D sketching, due to the challenges listed before, our problem is much more difficult compared to the existing VR/AR sketching systems requiring special equipment (e.g., HMDs and MoCap systems). It is expected that the quality of our results might not be comparable to those by the existing systems. However, our system has a clear advantage of more accessible and being able to use in both indoors (e-j) and outdoors (a-d,k,l), enabling new applications for example outdoor navigation, location-based real-world annotation.

8 CONCLUSIONS AND FUTURE WORK

In this work we took the first step to study the special challenges for in-context 3D sketching using a single AR-enabled mobile device, including the narrow field of view, the lack of stereo display, the coupling of the input and the output, etc. Relative drawing, various forms of snapping, and planar/curved surface proxies are carefully unified into a powerful workflow to alleviate the problems due to these challenges. We have shown how our proposed mobile AR 3D sketching interface, *Mobi3DSketch*, can benefit the creation of 3D sketches for early stages of creative design in the real world.

Our current implementation is highly dependent on the robustness of motion tracking of ARKit, which, however, is sometimes unstable and scene dependent. For tracking noise, it is possible to apply more advanced curve denoising methods to smooth out the noise. It would also be interesting to explore advanced drawing beautification methods [13].



(a) 23.3 mins (b) 7.0 mins (c) 11.6 mins

(d) 13.7 mins (e) 3.7 mins (f) 27.3 mins (g) 7.5 mins

(h) 12.4 mins (i) 13.5 mins (j) 9.3 mins (k) 8.4 mins (l) 17.5 mins

Other views

Figure 19: A gallery of 3D sketches created by *Mobi3DSketch*.

However, when the tracking is completely lost, we have nothing to do but improve the underlying tracking algorithm. In the future we are also interested in interactive tools for editing the drawn strokes, e.g., through handle-based feature-preservation curve deformation.

3D strokes themselves are not efficient to create very complicated 3D designs, since a lot of them are needed to create surface-like regions [2]. We found effective to allow filled planar and curve surface proxies to become parts of a design (see examples in Figure 19 (a, d, f, g)). In the future we will look into this direction further and even explore 3D surface modeling in mobile AR. Currently, we focused on designing a general-purpose 3D sketching system. Our system can be also tailor-made for specific applications and/or specific users (e.g., landscape architects). We believe our work can inspire further research along this line.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive comments and the study participants for their inputs. This work was partially supported by grants from HKSAR Research Grants Council (No. CityU 11300615, CityU 11204014), City University of Hong Kong (No. 7005176 (SCM)), the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Project No. VRLAB2018C11), and the Centre for Applied Computing and Interactive Media (ACIM) of SCM, City University of Hong Kong.

REFERENCES

[1] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *CHI '17 f.* 5643–5654.

[2] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. In *CHI '18*.

[3] Zane Assets. 2018. Paint Space AR. https://www.paintspacear.com/.

[4] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2009. EverybodyLovesSketch: 3D Sketching for a Broader Audience. In *UIST '09*. 59–68.

[5] Oriel Bergig, Nate Hagbi, Jihad El-Sana, and Mark Billinghurst. 2009. In-place 3D sketching for authoring and augmenting mechanical systems. In *ISMAR 2009*. 87–94.

[6] Eric A Bier. 1990. Snap-dragging in three dimensions. In *ACM SIGGRAPH Computer Graphics*, Vol. 24. 193–204.

[7] David Bourguignon, Marie-Paule Cani, and George Drettakis. 2001. Drawing for illustration and annotation in 3D. In *Computer Graphics Forum*, Vol. 20. Wiley Online Library, 114–123.

[8] Vinayak Cecil Piya. 2016. RealFusion: An Interactive Workflow for Repurposing Real-World Objects towards Early-stage Creative Ideation. In *Graphics Interface*.

[9] Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E Mackay, and Michel Beaudouin-Lafon. 2016. Beyond Snapping: Persistent, Tweakable Alignment and Distribution with StickyLines. In *UIST 2016*. ACM, 133–144.

[10] Michael F Deering. 1995. HoloSketch: a virtual reality sketching/animation tool. *ACM TOCHI* 2, 3 (1995), 220–238.

[11] Julie Dorsey, Songhua Xu, Gabe Smedresman, Holly Rushmeier, and Leonard McMillan. 2007. The mental canvas: A tool for conceptual architectural design and analysis. In *PG '07*. 201–210.

[12] Tomás Dorta, Gokce Kinayoglu, and Michael Hoffmann. 2016. Hyve-3D and the 3D Cursor: Architectural co-design with freedom in Virtual Reality. *International Journal of Architectural Computing* 14, 2 (2016), 87–102.

[13] Jakub Fišer, Paul Asente, Stephen Schiller, and Daniel Sỳkora. 2016. Advanced drawing beautification with ShipShape. *Computers & Graphics* 56 (2016), 46–58.

[14] Google. 2018. Just a Line. https://justaline.withgoogle.com/.

[15] Cindy Grimm and Pushkar Joshi. 2012. Just DrawIt: a 3D sketching system. In *SBIM '12*. Eurographics Association, 121–130.

[16] Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton. 2002. Creating Principal 3D Curves with Digital Tape Drawing. In *CHI '02*. 121–128.

[17] Ke Huo, Karthik Ramani, et al. 2017. Window-Shaping: 3D Design Ideation by Creating on, Borrowing from, and Looking at the Physical World. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 37–45.

[18] Bret Jackson and Daniel F Keefe. 2016. Lift-off: Using reference imagery and freehand sketching to create 3D models in VR. *IEEE TVCG* 22, 4 (2016), 1442–1451.

[19] Kiia Kallio. 2005. 3D6B editor: projective 3D sketching with line-based rendering. *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005).

[20] Daniel F Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H Laidlaw, and Joseph J LaViola Jr. 2001. CavePainting: a fully immersive 3D artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 85–93.

[21] Daniel F. Keefe, Robert C. Zeleznik, and David H. Laidlaw. 2007. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE TVCG* 13, 5 (2007), 1067–1081.

[22] Yongkwan Kim, Sang-Gyun An, Joon Hyub Lee, and Seok-Hyung Bae. 2018. Agile 3D Sketching with Air Scaffolding. In *CHI '18*. ACM, 238.

[23] Yongkwan Kim and Seok-Hyung Bae. 2016. Sketchingwithhands: 3D sketching handheld products with first-person hand posture. In *UIST '16*. ACM, 797–808.

[24] David Lakatos, Matthew Blackshaw, Alex Olwal, Zachary Barryte, Ken Perlin, and Hiroshi Ishii. 2014. T (ether): spatially-aware handhelds, gestures and proprioception for multi-user 3D modeling and animation. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 90–93.

[25] Mingyang Li and Anastasios I Mourikis. 2013. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research* 32, 6 (2013), 690–711.

[26] Yuwei Li, Xin Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. 2017. SweepCanvas: Sketch-based 3D prototyping on an RGB-D image. In *UIST 2017*.

[27] Hod Lipson and Moshe Shpitalni. 1996. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design* 28, 8 (1996), 651–663.

[28] Hengmao Liu. 2018. AR+Brush. https://itunes.apple.com/nz/app/ar-brush/id1273579764?mt=8.

[29] Andrew YC Nee, SK Ong, George Chryssolouris, and Dimitris Mourtzis. 2012. Augmented reality applications in design and manufacturing. *CIRP Annals-manufacturing technology* 61, 2 (2012), 657–679.

[30] Esha Nerurkar, Simon Lynen, and Sheng Zhao. 2017. System and method for concurrent odometry and mapping. US Patent App. 15/595,617.

[31] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D Wilson. 2016. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1233–1244.

[32] Patrick Paczkowski, Min H. Kim, Yann Morvan, Julie Dorsey, Holly Rushmeier, and Carol O'Sullivan. 2011. Insitu: Sketching Architectural Designs in Context. *ACM Trans. Graph.* 30, 6, Article 182 (2011), 10 pages.

[33] Devarajan Ramanujan, Cecil Piya, Karthik Ramani, et al. 2016. MobiSweep: Exploring spatial design ideation using a smartphone as a hand-held reference plane. In *TEI'16*. 12–20.

[34] Emanuel Sachs, Andrew Roberts, and David Stoops. 1991. 3-Draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications* 6 (1991), 18–26.

[35] Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *CHI '01*. ACM, 261–268.

[36] Ryan Schmidt, Azam Khan, Karan Singh, and Gord Kurtenbach. 2009. Analytic Drawing of 3D Scaffolds. *ACM Transactions on Graphics* 28, 5 (2009). Proceedings of SIGGRAPH ASIA 2009.

[37] Katie A Siek, Yvonne Rogers, and Kay H Connelly. 2005. Fat finger worries: how older and younger users physically interact with PDAs. In *IFIP Conference on Human-Computer Interaction*. Springer, 267–280.

[38] Shunichi Tano, Shinya Yamamoto, Junko Ichino, Tomonori Hashiyama, and Mitsuru Iwata. 2013. Truly Useful 3D Drawing System for Professional Designer by "Life-sized and Operable" Feature and New Interaction. In *INTERACT 2013*. Springer, 37–55.

[39] Min Xin, Ehud Sharlin, and Mario Costa Sousa. 2008. Napkin Sketch: Handheld Mixed Reality 3D Sketching. In *VRST '08*. 223–226.

[40] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. *ACM Trans. Graph.* 33, 4, Article 131 (2014), 13 pages.

[41] Pengfei Xu, Hongbo Fu, Youyi Zheng, Karan Singh, Hui Huang, and Chiew-Lan Tai. 2018. Model-guided 3D sketching. *IEEE Transactions on Visualization and Computer Graphics* (2018).

[42] Brandon Yee, Yuan Ning, and Hod Lipson. 2009. Augmented reality in-situ 3D sketching of physical objects. In *Intelligent UI workshop on sketch recognition*.

[43] Youyi Zheng, Han Liu, Julie Dorsey, and Niloy J. Mitra. 2016. SmartCanvas: Context-inferred Interpretation of Sketches for Preparatory Design Studies. *Comput. Graph. Forum* 35, 2 (2016), 37–48.

[44] Changqing Zou, Shifeng Chen, Hongbo Fu, and Jianzhuang Liu. 2015. Progressive 3D reconstruction of planar-faced manifold objects with DRF-based line drawing decomposition. *IEEE TVCG* 21, 2 (2015), 252 – 263.